

# FRAMEWORK SYSTEM

Publication number: WO02054263 (A1)

Publication date: 2002-07-11

Inventor(s): ISHIBASHI KUNIHITO [JP]; MAESHIMA MITSURU [JP]; OKUMURA NARIHIRO [JP]; SAKASHITA ISAO [JP]; IGAKURA YOKO [JP] +

Applicant(s): FUTURE SYSTEM CONSULTING CORP [JP]; ISHIBASHI KUNIHITO [JP]; MAESHIMA MITSURU [JP]; OKUMURA NARIHIRO [JP]; SAKASHITA ISAO [JP]; IGAKURA YOKO [JP] +

Classification:

- International: G06F9/46; G06Q10/00; H04L12/58; G06F9/46; G06Q10/00; H04L12/58; (IPC1-7): G06F15/00; G06F15/16

- European: H04L12/58; G06F9/46R6M; G06Q10/00F

Application number: WO2001JP11532 20011227

Priority number(s): JP20000401794 20001228

Also published as:

EP1347390 (A1)  
US2007130247 (A1)  
US7366751 (B2)  
US2003014551 (A1)  
US7177899 (B2)



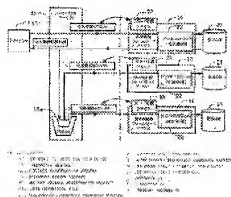
more >>

Cited documents:

JP2000250768 (A)  
JP1253036 (A)  
JP2113362 (A)  
JP11015796 (A)  
EP1024429 (A2)

## Abstract of WO 02054263 (A1)

A messaging service (15) relays request messages including a subject ID from clients (11, 13) to a framework service (16). The framework service (16) has business logics (22) and flow definition files (23). Each flow definition file (23) contains definition sentences corresponding to various subject IDs. Each sentence describes an execution schedule of a business logic. The framework service (16), according to the execution schedule corresponding to the subject ID of a received message, selects one or more business logics and executes them. The flow definition files (23) can be updated at any desired time even in operation. If some of the business logics are cooperatively executed, a business logic succeeding the preceding business logic can be executed synchronously or asynchronously depending on the description of the flow definition file.



Data supplied from the **espacenet** database — Worldwide





(74) 代理人: 上村 輝之, 外 (KAMIMURA, Teruyuki et al.); 〒130-0022 東京都 墨田区 江東橋 1 丁目 8 番 3-702 号 Tokyo (JP). 添付公開書類: 国際調査報告書

(81) 指定国 (国内): CN, IN, JP, KR, US.

(84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

2 文字コード及び他の略語については、定期発行される各 PCT ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

---

(57) 要約:

メッセージングサービス 15 は、クライアント 11, 13 とフレームワークサービス 16 の間で、サブジェクト ID を含むリクエストメッセージを中継する。フレームワークサービス 16 は、複数のビジネスロジック 22 とフロー定義ファイル 23 を有する。フロー定義ファイル 23 には、様々なサブジェクト ID に対応した複数の定義センテンスが含まれ、各定義センテンスには、ビジネスロジックの実行スケジュールが記述されている。フレームワークサービス 16 は、受け取ったメッセージのサブジェクト ID に対応する実行スケジュールに従って、1 又は複数のビジネスロジックを選び実行する。フロー定義ファイル 23 は、稼動中であっても所望の時に更新できる。複数のビジネスロジックを連係させて実行する場合、フロー定義ファイルの記述次第で、先行するビジネスロジックに対して後続のビジネスロジックを同期的に実行することも非同期的に実行することもできる。

## 明 細 書

### フレームワークシステム

#### 技 術 分 野

本発明は、クライアントからのメッセージに応答して、選択された 1 又はそれ以上のビジネスロジックを実行するフレームワークシステムに関する。

#### 技 術 背 景

この種のフレームワークシステムは、典型的には、1 又はそれ以上のクライアントと通信可能に接続される 1 又はそれ以上のサーバによって構成される。この種のフレームワークシステムには、従来からメッセージングサービスと、フレームワークサービスという 2 つの基本機能が提供されている。メッセージングサービスは、クライアントとサーバ間及びサーバとサーバ間のメッセージ通信を処理するサービスである。フレームワークサービスは、通常複数のビジネスロジック（業務処理プロセス）を有しており、クライアントからのメッセージに応答して、選択された 1 又はそれ以上のビジネスロジックを同期方式で又は非同期方式で実行することによりトランザクションを遂行する。

しかしながら、上記従来例にあつては、フレームワークサービスにおけるビジネスロジックの処理スケジュールが固定的にプログラムされている。そのため、多種多様なメッセージに応じて複雑なビジネスロジックのフローを柔軟に定義することが困難である。

また、クライアントからのメッセージに応答して、複数のビジネスロジックを逐次に実行する場合がある。例えば、クライアントからの仕入登録

リクエストに応答して、仕入登録や在庫更新や買掛金記帳などを行う場合である。このような場合、従来システムでは、例えば仕入登録のようなフロントデスクに関わるビジネスロジックはそのメッセージに同期して実行するが、在庫更新や買掛金記帳のようなバックオフィスに関わるビジネスロジックは夜間のパッチ処理により実行する。そのため、バックオフィスの処理の結果は、翌日まで待たないと参照できない。しかし、新たな取引のチャンスをいち早くキャッチするには、バックオフィスのビジネスロジックも可能な限り早期に実行してその結果が早期に参照できることが望ましい。

また、複数のフレームワークサーバを備えたシステムでは、フレームワークサーバ相互間でもメッセージを送受しながら協働して、一連のビジネスロジックを実行していく。各サーバのステータス（例えば、保有するビジネスロジックの種類、正常に稼働しているか否か、ビジーか否か、など）は、他のサーバとは違うのが普通であり、且つ、時間経過に伴って変化もする。このことは、ビジネスロジックの実行をどのサーバに依頼するかを決めるスケジューリングを複雑にする。

## 発 明 の 開 示

本発明は、ビジネスロジックフローを、プログラムすることなく、定義し変更することが容易なフレームワークシステムを提供することを目的とする。

本発明の別の目的は、一連のビジネスロジックを、完全にリアルタイムではなくても、できるだけリアルタイムに近い態様で、実行することができるフレームワークシステムを提供することにある。

本発明の更に別の目的は、複数のフレームワークサーバを備えたシステ

ムにおいて、一連のビジネスロジックのスケジューリングを、それらサーバのステータスに応じて適切に行えるようにすることにある。

本発明の一つの観点に従う、クライアントと通信可能に接続されるフレームワークシステムは、複数のビジネスロジックと関連付けられ、クライアントからのリクエストのメッセージに回答して、選択された1又はそれ以上のビジネスロジックを実行し、そして、クライアントへのリプライのメッセージを出力するフレームワークサービスと；クライアントとフレームワークサービスとの間に介在し、クライアントとフレームワークサービスとの間でメッセージを中継するメッセージングサービスと；フレームワークサービスに関連付けられたフロー定義ファイルとを備える。リクエストのメッセージは、そのメッセージのサブジェクトに関するサブジェクトIDを含んでいる。フロー定義ファイルは、複数の異なるサブジェクトIDにそれぞれ対応した複数の定義センテンスを含んでおり、各定義センテンスは所定の1又はそれ以上のビジネスロジックのための実行スケジュールを記述している。フレームワークサービスは、メッセージングサービスからリクエストのメッセージを受け取ると、定義ファイル内の受信されたメッセージのサブジェクトIDに対応した定義センテンスを参照し、参照された定義センテンスに記述された実行スケジュールに従って、実行すべき1又はそれ以上のビジネスロジックを選択する。

好適な実施形態では、フレームワークシステムは、フレームワークサービスが稼動している間に、フロー定義ファイルを更新するフロー定義更新コンポーネントを更に備える。

好適な実施形態では、また、フロー定義ファイルには、一つの先行のビジネスロジックと少なくとも一つの後続のビジネスロジックとを含む複数のビジネスロジックの連係された実行のための連係実行スケジュール

を記述した定義センテンスを含むことができる(勿論、連係実行スケジュールを定義した定義センテンスを含まなくてもよい)。その連係実行スケジュールは、後続のビジネスロジックの実行に関して同期方式と非同期方式の選択に関する連係モードを含んでいる。そして、フレームワークサービスは、連係実行スケジュールに従って複数のビジネスロジックを実行する場合、連係実行スケジュール内の連係モードに従って、先行のビジネスロジックの実行と同期的に又は非同期的に、後続のビジネスロジックを実行する。

好適な実施形態では、連係実行スケジュールは、先行のビジネスロジックに関する情報と、後続のビジネスロジックのための第2のリクエストのメッセージに関する情報とを含む。そして、フレームワークシステムは、連係実行スケジュールに従って複数のビジネスロジックを実行する場合、先行のビジネスロジックを実行した後、先行のビジネスロジックの実行結果を用いて第2のリクエストのメッセージを生成して、その第2のリクエストのメッセージをメッセージングサービスに送る。その後、フレームワークシステムは、メッセージングサービスから上記第2のリクエストのメッセージを受けると、その第2のリクエストのメッセージに対応する定義センテンスの実行スケジュールに従って、後続のビジネスロジックを実行する。

好適な実施形態では、また、メッセージングサービスが、前記第2のリクエストのメッセージを保証する為の不揮発性のメッセージキューを備える。このメッセージキューは、第2のリクエストメッセージがフレームワークサービスへ送られた後も、その第2のリクエストメッセージを消さずに保存する。そして、メッセージングサービスは、その不揮発性のメッセージキューで保存されている第2のリクエストのメッセージを、必要に応

じて、フレームワークシステムへ再送信することができる。

本発明の別の観点に従うクライアントと通信可能に接続されるフレームワークシステムは、クライアントからのリクエストのメッセージを処理し、そして、クライアントへのリプライのメッセージを出力するフレームワークサービスと；クライアントとフレームワークサービスとの間に介在し、クライアントとフレームワークサービスとの間でメッセージを中継するメッセージングサービスとを備える。そして、メッセージングサービスは、上記リクエストとリプライのメッセージのような P to P 通信のメッセージだけでなく、クライアントとフレームワークサービスとの間の P to M 通信のメッセージをも中継する。

好適な実施形態では、メッセージングサービスが、P to M 通信のメッセージを一時的に待たせるためのリングバッファを有する。

好適な実施形態では、また、P to P 通信のメッセージの各々には、メッセージ保証の要否に関する保証モードが含まれている。そして、メッセージングサービスは、メッセージ保証が不要な P to P 通信のメッセージを一時的に待たせるための第 1 のメッセージキューと、メッセージ保証が必要な P to P 通信のメッセージを一時的に待たせるための不揮発性の第 2 のメッセージキューとを有する。

本発明のまた別の観点に従う、クライアントと通信可能に接続されるフレームワークシステムは、クライアントと接続された少なくとも一つのメッセージングサービスを含んだ、相互にメッシュ接続されている複数のメッセージングサービスと；複数のメッセージングサービスにそれぞれ接続された複数のフレームワークサービスとを備える。そして、複数のフレームワークサービスの各々は、予め定められたサブジェクト ID をもったリクエストのメッセージを待ち受け、その予め定められたサブジェクト ID



をもったリクエストのメッセージを受信すると、受信されたリクエストのメッセージを処理し、そして、クライアントへのリプライのメッセージを出力することができる。そして、上記複数のメッセージングサービスは、クライアントと上記複数のフレームワークサービスの間でのメッセージの中継、及び、上記複数のフレームワークシステム相互間でのメッセージの中継を行う。

好適な実施形態では、各メッセージングサービスが、他の1又はそれ以上のメッセージングサービスの稼働を監視し、他の或るメッセージングサービスについて正常な稼働を検出できなかった場合は、その正常に稼働していないメッセージングサービスへメッセージを中継する代わりに、他の正常に稼働しているメッセージングサービスへメッセージを中継する。

好適な実施形態では、また、各メッセージングサービスが、各々に接続されたフレームワークサービスの状態を監視し、監視された状態に応じて、クライアントからのリクエストのメッセージを上記フレームワークサービスに中継するか、他の何れかのメッセージングサービスに中継するかを、選択する。

好適な実施形態では、また、各メッセージングサービスは独自の管理テーブルを有し、各管理テーブルには、各メッセージングサービスに接続されたフレームワークシステムが待ち受けるメッセージのサブジェクト ID、及び、各メッセージングサービスに接続された他の1又はそれ以上のメッセージングサービスがそれぞれ待ち受けるメッセージのサブジェクト ID が登録されている。そして、各メッセージングサービスは、メッセージを受信したとき、それ独自の管理テーブルを参照して、受信されたメッセージのサブジェクト ID にマッチするサブジェクト ID を待ち受けるフレームワークサービス又は他のメッセージングサービスを探し、見つかったフレ

ームワークサービス又は他のメッセージングサービスへ、受信されたメッセージを中継する。

好適な実施形態では、さらに、各メッセージングサービスは、それ独自の管理テーブルの内容に基づいて、そのメッセージングサービスが待ち受けるメッセージのサブジェクト ID を、他の 1 又はそれ以上のメッセージングサービスに通知する。その通知を受けた他のメッセージングサービスの各々は、通知された内容が、それ独自の管理テーブルに未だ登録されていないければ、その通知内容を、それ独自の管理テーブルに追加登録する。

本発明の更に別の観点に従う、クライアントと通信可能に接続されるフレームワークシステムは、クライアントからのリクエストのメッセージを処理し、そして、クライアントへのリプライのメッセージを出力するフレームワークサービスと；クライアントとフレームワークサービスとの間に介在し、クライアントとフレームワークサービスとの間でメッセージを中継するメッセージングサービスとを備える。クライアントからのリクエストのメッセージは所与の優先順位を有している。そして、メッセージングサービスは、上記リクエストのメッセージを一時的に待たせるためのメッセージキューと、メッセージキューの入出力を管理するキュー管理コンポーネントとを備える。キュー管理コンポーネントは、メッセージキューに複数のメッセージが蓄積されているとき、各メッセージの優先順位に応じて、複数のメッセージのメッセージキューからの出力順序を制御する優先モードと；メッセージキューに複数のメッセージが蓄積されているとき、メッセージキューから先に取り出されたメッセージの処理が前記フレームワークサービスにおいて完了するまで、メッセージキューに蓄積されている他のメッセージの取り出しを禁止するシーケンス保証モードとを備える。

## 図面の簡単な説明

図 1 は、本実施形態の全体構成図である。

図 2 は、RtFA サーバ 1 4 が取り扱うメッセージの構成を示すブロック図である。

図 3 は、RtFA サーバ 1 4 の構成図である。

図 4 は、メッセージングサービス 1 5 に関係付けられる管理テーブル 1 8 の例を示す。

図 5 は、複数の RtFA サーバ間の接続構成図である。

図 6 は、複数のメッセージングサービス間の接続構成図である。

図 7 は、P to M (1 対多) 通信処理のメッセージ中継方法を示す説明図である。

図 8 は、P to M 通信処理の一例を示すフローチャートである。

図 9 は、P to P (1 対 1) 通信処理を示すフローチャートである。

図 1 0 は、メッセージングサービスのロードバランシング機能の説明図である。

図 1 1 は、フレームワークサービスの構成を示すブロック図である。

図 1 2 は、フレームワークサービスの動作説明図である。

図 1 3 は、フロー定義ファイルの簡単な例を示す。

図 1 4 は、フロー定義ファイルの更新システムを示すブロック図である。

図 1 5 は、フレームワークサービスのメッセージ変換処理機能を示すブロック図である。

図 1 6 は、メッセージングサービスのメッセージキューの構成を示すブロック図である。

図 1 7 は、メッセージキュー管理機能がメッセージキューにメッセージ

を格納する動作を示すフローチャートである。

図 18 は、図 17 のフローチャートの続きである。

図 19 は、メッセージキュー管理機能のメッセージ読出機能を示すブロック図である。

図 20 は、優先モードのための記憶域の構成を示す。

図 21 は、優先モードの動作の流れを示す。

図 22 は、シーケンス保証モードの動作の流れを示す。

図 23 は、フレームワークサービスのメッセージ判別機能を示すブロック図である。

図 24 は、フレームワークサービスのメッセージ変換ビジネスロジックを示すブロック図である。

#### 発明を実施するための最良の形態

以下、本発明の一実施形態を図面を参照して説明する。図 1 は、本実施形態の全体構成を示す。

本システムは、プレゼンテーション層 1、ビジネスロジック層 2、及びデータサービス層 3 から構成されている。

プレゼンテーション層 1 は、ウェブ (World Wide Web) クライアント 11、ウェブ (World Wide Web) サーバ 12、又は VB (Visual Basic) クライアント 13 を含んでいる。

ビジネスロジック層 2 には、本発明の原理に従うフレームワークアーキテクチャ (この明細書では「RtFA: Real Time Framework Architecture」という) が適用されたサーバコンピュータシステムである RtFA サーバ 14 が設けられている。RtFA サーバ 14 は、メッセージングサービスを行うサーバコンピュータシステム (以下、簡単に「メッセージングサービス」

という) 15と、フレームワークサービスを行うサーバコンピュータシステム(以下、簡単に「フレームワークサービス」という) 16とを備えている。典型的には、メッセージングサービス15とフレームワークサービス16は、それぞれ、別のコンピュータマシン又は別のコンピュータマシンのセットを用いて具現化される。しかし、必ずしもそうでなければならぬわけではなく、同一のコンピュータマシン又はコンピュータマシンのセットを用いて、メッセージングサービス15とフレームワークサービス16の双方がインプレメントされることも可能である。

メッセージングサービス15は、クライアント11、13とRtFAサーバ14間のメッセージのやり取りを制御する。すなわち、メッセージングサービス15は、クライアント11、13から、或る処理のためのリクエストのメッセージを受信し、そのリクエストのメッセージを、リクエストされた処理を行なうことができるフレームワークサービス16に渡す。また、メッセージングサービス15は、フレームワークサービス16から、実行された処理の結果を示すリプライのメッセージを受信し、更に、場合によっては、後続の追加の処理のための追加のリクエストのメッセージも受信する。そして、メッセージングサービス15は、そのリプライのメッセージをクライアント11、13に送ったり、追加のリクエストのメッセージをフレームワークサービス16に再び渡したりする。

メッセージングサービス15によって取り扱われる全てのメッセージの各々には、そのメッセージのサブジェクト(例えば、仕入登録リクエスト、在庫更新リクエスト、買掛登録リクエストなど)を示す識別コードであるサブジェクトIDが含まれている。このサブジェクトIDは、後述するように、メッセージングサービス15又はフレームワークサービス16によって、そのメッセージの宛先を選択する目的に利用され、そのため、「宛

先サブジェクト ID」とも呼ばれる。

フレームワークサービス 16 は、様々な種類のメッセージをそれぞれ処理するための複数のビジネスロジック (例えば、仕入処理ビジネスロジック、在庫処理ビジネスロジック、買掛処理ビジネスロジック、メッセージ変換ビジネスロジックなど) を有している。フレームワークサービス 16 は、メッセージングサービス 15 から或るメッセージを受け取ると、そのメッセージのサブジェクト ID からそのメッセージを処理するためのビジネスロジックを選び、その選んだビジネスロジックを呼び出しす。呼び出されたビジネスロジックは、そのメッセージの内容に応じた処理を行い、データサービス層 9 のデータを更新する。メッセージングサービス 15 及びフレームワークサービス 16 のためのコンピュータプログラムは、例えば、JAVA 言語 (商標) によって記述されている。

データサービス層 3 は、データベースサーバ (DB サーバ) 17 を備えている。

ウェブクライアント 11 とウェブサーバ 12 の間には HTTP 又はソケットによる通信コネクションが確立され得る。ウェブクライアント 11 と RtFA サーバ 14 との間には、ソケットによる直接的な通信コネクションが確立され得る。VB クライアント 13 と RtFA サーバ 14 との間には、ソケットによる直接的な通信コネクションが確立され得る。また、ウェブサーバ 12 と RtFA サーバ 14 との間には、ソケットによる直接的な通信コネクションが確立され得る。更に、RtFA サーバ 14 と DB サーバ 17 との間には JDBC による通信コネクションが確立され得る。

図 2 は、RtFA サーバ 14 が取り扱うメッセージの構成を示す。

図 2 に示すように、メッセージ 10 は、複数のサーバ制御項目と業務データとから構成される。サーバ制御項目は、RtFA サーバ 14 を制御するた

めの各種のデータであり、例えば、メッセージ種別、保証モード、優先順位、サブジェクト ID (宛先サブジェクト ID) 及び送信元情報などが含まれている。

「メッセージ種別」は、そのメッセージが P to P (Point to Point) 通信のメッセージであるか、P to M (Point to Many) 通信のメッセージであるかを区別する。ここで、P to P 通信とは、或る 1 つのコンピュータシステムから別の 1 つのコンピュータシステムへとメッセージが送られる通信、つまり、1 対 1 通信のことをいう。P to P 通信の一例は、或るクライアントからの仕入登録リクエストのメッセージを、仕入処理を実行する或る一つのフレームワークサービスへ送る場合である。一方、P to M 通信とは、或る 1 つ又は複数のコンピュータシステムから別の複数のコンピュータシステムへとメッセージが送られる通信、つまり、1 対複数又は複数対複数の通信のことをいう。P to M 通信の一例は、或る 1 つのフレームワークサービスからの在庫更新結果を示すメッセージを、フロントデスクの仕事を担う複数のクライアント端末へ送信する場合である。

「保証モード」は、メッセージ保証 (すなわち、そのメッセージが確実に送信及び処理されることの保証) をする (“G”) か否か (“N”) を区別する。保証モードが “N” のメッセージを受信したとき、メッセージングサービス 15 は、そのメッセージを、揮発性の半導体メモリを用いたメッセージキュー (以下、「メモリキュー」という) に入れる。一方、保証モードが “G” のメッセージを受信したとき、メッセージングサービス 15 は、そのメッセージを、不揮発性のディスクストレージを用いたメッセージキュー (以下、「DB (データベース) キュー」という) に入れ、そのメッセージを送信した後も、そのメッセージを DB キューで保存しておく。従って、保証モードが “G” のメッセージについては、送信失敗やシステ

ムダウンなどの問題が生じたとしても、後に再度、同じメッセージを送信することができる。

「優先順位」は、そのメッセージの処理の優先レベルを示す。例えば、ハイ（“H”）、ノーマル（“N”）及びロー（“L”）の3種類の優先順位がある。後述するように、メッセージングサービス15は、メッセージキュー内で複数のメッセージが待っているとき、それらのメッセージをメッセージキューから取り出す順序を、それらのメッセージのそれぞれの優先順位に応じて制御することができる。

「サブジェクトID（宛先サブジェクトID）」は、既に述べたとおり、そのメッセージのサブジェクトの種別を示す識別コードである。後述するように、メッセージングサービス15は、メッセージを受信すると、そのメッセージのサブジェクトIDに基づいて、そのメッセージの宛先となるべきコンピュータシステム（例えば、クライアント、フレームワークシステム、他のメッセージングサービスなど）を選択する。また、フレームワークサービス16は、メッセージを受信すると、そのメッセージのサブジェクトIDに基づいて、そのメッセージを処理するために呼び出すべきビジネスロジックを選択する。

「送信元情報」は、そのメッセージの送信元のサブジェクトやIPアドレスやコンピュータ名などを表す。

図3は、RtFAサーバ14の構成図である。

既に参照した図1では、1つのRtFAサーバ14しか示されていないかった。しかし、実際には、図3に示すように、複数のRtFAサーバ14を設けることができる。図3に示す例では、2つのRtFAサーバ14が設けられているが、もっと多くのRtFAサーバ14を設けることができる。

図3に示すように、各RtFAサーバ14には、一つのメッセージングサ



ービス 15 と、1 又は複数のフレームワークサービス 16 が設けられ、メッセージングサービス 15 とフレームワークサービス 16 は相互に通信可能に接続されている。メッセージングサービス 15 は、また、1 又は複数のクライアント 11、13 と通信可能に接続されている。メッセージングサービス 15 は、クライアント 11、13 及びフレームワークサービス 16 とそれぞれ通信する複数のスレッド 171 を並列に処理している。

複数の RtFA サーバ 14 が存在するので、メッセージングサービス 15 とフレームワークサービス 16 の組は複数組存在している。そして、複数の RtFA サーバ 14 のメッセージングサービス 15 は、相互に通信可能に接続されている。各 RtFA サーバ 14 のメッセージングサービス 15 は、他の RtFA サーバ 14 のメッセージングサービス 15 との通信を処理するスレッド 171 も備えている。各メッセージングサービス 15 が持つ複数の通信スレッド 171 は、それぞれキープアライブ機能を有し、この機能により、そのメッセージングサービス 15 にコネクションされたクライアント 11、13、他のメッセージングサービス 15、及びフレームワークサービス 16 と相互の正常な稼働を確認しあっている。

また、各メッセージングサービス 15 には、図 4 に例示するような管理テーブル 18 が関連付けられている。管理テーブル 18 には、そのメッセージングサービス 15 が伝達すべきメッセージの経路、換言すれば、そのメッセージングサービス 15 が取り扱う様々なメッセージの送り先、が管理される。図 4 に示すように、管理テーブル 18 には、そのメッセージングサービス 15 に接続された全てのコンピュータシステム（クライアント 11、13、他のメッセージングサービス 15、及びフレームワークサービス 16）の各々について、種別、コンピュータ ID、IP アドレス、P to P サブジェクト ID、P to M サブジェクト ID 及び処理中フラグなどが登録さ

れる。

「種別」は、そのコンピュータシステムの種別、例えば、フレームワークサービス (F)、メッセージングサービス (M)、クライアント端末ターミナル (T) の区別を示す。

「コンピュータ ID」及び「IP アドレス」は、そのコンピュータシステムの固有の識別コードと IP アドレスを示す。

「P to P サブジェクト ID」は、そのコンピュータシステムが待ち受ける (つまり、そのコンピュータシステムが、そのメッセージの宛先になり得る) P to P 通信用のメッセージのサブジェクト ID (宛先サブジェクト ID) を示す。各コンピュータシステムごとに、複数の P to P サブジェクト ID を登録することができる。

「P to M サブジェクト ID」は、そのコンピュータシステムが待ち受ける (つまり、そのコンピュータシステムが、そのメッセージの宛先になり得る) P to M 通信用のメッセージのサブジェクト ID (宛先サブジェクト ID) を示す。各コンピュータシステムごとに、複数の P to M サブジェクト ID を登録することができる。

「処理中フラグ」は、そのコンピュータシステムとの通信のためのスレッド 171 が、P to P 通信の処理中である (「1」) か否 (「0」) かを示す。

図 4 に示した管理テーブル 18 の登録内容は、図 3 中の左側のメッセージングサービス Mes1 の場合の例であり、そこには、左側のメッセージングサービス Mes1 に接続された全てのコンピュータシステム、すなわち、2つのフレームワークサービス Frn1、Frn2、右側のメッセージングサービス Mes2 及び 3つのクライアント端末 PC1、PC2 及び PC3 が登録されている。この例によれば、1 番目のフレームワークサービス Frn1 は、P to P 通信

に関しては“ A ”、“ B ”、“ C ”及び“ D ”というサブジェクト ID をそれぞれもった4種類のメッセージを待ち受けており、また、P to M 通信に関しては“ 0 ”、“ P ”、“ Q ”及び“ R ”というサブジェクト ID をそれぞれもった4種類のメッセージを待ち受けている。また、図中右側のメッセージングサービス Mes2 は、左側のメッセージングサービス Mes1 からの P to P 通信のメッセージとして、“ D ”、“ E ”、“ F ”、“ H ”、“ I ”及び“ J ”というサブジェクト ID をそれぞれもった7種類のメッセージを待ち受けており、また、左側のメッセージングサービス Mes1 からの P to M 通信のメッセージとして“ P ”、“ Q ”、“ R ”、“ S ”、“ T ”及び“ U ”というサブジェクト ID をそれぞれもった6種類のメッセージを待ち受けている。

再び図3を参照して、各メッセージングサービス15は、メッセージを受信すると、そのメッセージがP to P 通信のものかP to M 通信のものを判断する。P to P 通信のメッセージを受信したときは、メッセージングサービス15は、そのメッセージングサービス15に関連付けられている図4に例示したような管理テーブル18を参照して、その受信メッセージのサブジェクト ID とマッチするP to P サブジェクト ID のメッセージを待ち受けているコンピュータシステムを探し、見つかった1以上のコンピュータシステムの中から処理中フラグが“ 0 ”である1つのコンピュータシステムを選び、そして、選ばれた1つのコンピュータシステムにその受信メッセージを送信する。また、P to M 通信のメッセージを受信したときは、メッセージングサービス15は、そのメッセージングサービス15に関連付けられている図4に示したような管理テーブル18を参照して、その受信メッセージのサブジェクト ID とマッチするP to M サブジェクト ID のメッセージを待ち受けるコンピュータシステムを全て選び、そして、選ばれた全てのコンピュータシステムにそのメッセージを送信す

る。

図5は、複数の RtFA サーバ間の接続構成図である。図6は、複数の RtFA サーバ内のメッセージングサービスとフレームワークサービスの接続構成図である。

図5の上段又は下段に示すように、各 RtFA サーバ14は他の全ての RtFA サーバ14と相互に通信可能に接続されている。すなわち、図6に示すように、各 RtFA サーバ14のメッセージングサービス15が、他の全ての RtFA サーバ14のメッセージングサービス15と相互に通信可能に接続されている。このように複数のコンポーネントの各々が他の全てのコンポーネントと通信可能に接続されることを「メッシュ接続」という。また、図6に示すように、各メッセージングサービス15に対しては複数のフレームワークサービス16を直接連絡させることも可能である。

このように複数のメッセージングサービス15のメッシュ接続の下で、各メッセージサービス15は、所定の時期、例えばその起動時に、上述したキープアライブ機能を実行する。すなわち、各メッセージングサービス15は、起動すると、自分の管理テーブル18に既に登録されている他のメッセージングサービスの全てと逐次に接続して、自分と接続されているフレームワークシステム16及びクライアント11、13が待ち受けるメッセージのサブジェクト ID を、接続された他のメッセージングサービスの全てに通知する。通知されたサブジェクト ID は、そのメッセージングサービス15が他のメッセージングサービスから待ち受けるメッセージのサブジェクト ID として、他のメッセージングサービスの管理テーブルに登録される。また、各メッセージングサービス15は、他のメッセージングサービスと接続しようとしたときに接続できなかった場合には、その後一定間隔で複数回接続をリトライする。また、各メッセージサービス

15は、自分の管理テーブル18に未だ登録されていない他のメッセージングサービスから接続要求を受けたときには、その未登録のメッセージングサービスと接続して、未登録のメッセージングサービスとそれぞれが待ち受けるサブジェクトIDなどを交換し合い、そして、そのメッセージングサービスを自分の管理テーブル18に登録する。このようなキープアライブ機能のおかげで、特定のメッセージングシステム15が正常に稼働していない(例えば、システムダウンした)場合には、他の正常に稼働している各メッセージングサービス15は、その正常に稼働していない特定のメッセージングシステム15をメッセージの中継経路から除外し、自分の管理テーブル18を参照して、自動的に別の中継経路を選択することができる。

次に、図7乃至図9に基づいて、メッセージングサービス15が提供する通信サービスの種類を説明する。メッセージングサービス15が提供するデータ配信サービスには、P to M (1対多) 通信処理と、P to P (1対1) 通信処理の2種類がある。

P to M 通信処理は、前述したように、図4に例示したような管理テーブル18に登録されている各コンピュータシステムのP to M サブジェクトIDに基づいて、メッセージングサービス15が、受信したP to M メッセージを、それを待ち受ける全てのコンピュータシステムに配信する処理である。例えば、マーケット情報が更新される都度、更新されたマーケット情報のメッセージを、1乃至n個のRtFAサーバからn個のクライアントに配信する場合に、P to M 通信処理が行われる。

一方、P to P 通信処理は、前述したように、図4に例示したような管理テーブル18に登録されている各コンピュータシステムのP to P サブジェクトIDに基づいて、メッセージングサービス15が、受信したP to P メッセージを、それを待ち受けるコンピュータシステムの中の一つに送信

する処理である。例えば、クライアントからの或る処理のためのリクエストのメッセージを、その処理を行うビジネスロジックを有した一つのフレームワークサービスに送る場合に、P to P 通信処理が行われる。

図 7 は、P to M 通信処理のメッセージ中継方法の一例を示している。パブリッシャーとしての或るフレームワークサービス 16 から出力された或る P to M メッセージは、そのメッセージを待ち受ける全てのメッセージングサービス 15 に配信され、そして、それらのメッセージングサービス 15 から、そのメッセージを待ち受ける全てのクライアント 11、13 に配信される。

図 8 は、P to M 通信処理の一例を示すフローチャートである。まず、サブスクライバの一つであるクライアント T1 は、所定のメッセージングサービス Mes1 との間で、ソケットによるコネクションを確立する。また、クライアント T1 から当該メッセージングサービス Mes1 に対し自己が待ち受ける P to M メッセージのサブジェクト ID “X” を登録する。当該メッセージングサービス Mes1 は、自己に関連付けられた管理テーブル 18 に、クライアント T1 の P to M サブジェクト ID “X” を追加する。一方、クライアント T1 は、メッセージングサービス Mes1 からのメッセージの配信を待つ。その後、メッセージングサービス Mes1 に接続する他のメッセージングサービス Mes2, Mes3 が起動したとき、メッセージングサービス Mes1 の管理テーブル 18 に既に登録されたクライアント T1 の P to M サブジェクト ID “X” は、他のメッセージングサービス Mes2 及び Mes3 にも通知されて、それらの管理テーブル 18 に、メッセージングサービス Mes1 の P to M サブジェクト ID として “X” が追加される。

ここで、パブリッシャーとしてのフレームワークサービス Frm3 からサブジェクト ID “X” をもつ P to M メッセージが配信されると、これを受

けたメッセージングサービス Mes3 は、自己に関連付けられた管理テーブル 18 を参照し、サブジェクト ID “X” を待ち受ける全てのメッセージングサービス Mes1 及び Mes2 をピックアップし、それらに当該メッセージを引き渡す。

クライアント T 1 とコネクタされたメッセージングサービス Mes1 は、当該メッセージを受信すると、自己に関連付けられた管理テーブル 18 からサブジェクト ID “X” を待ち受けるクライアント T 1 をピックアップし、当該メッセージをそのクライアント T 1 に配信する。

図示していないが、各フレームワークサービスも、そのフレームワークサービスと直接連絡可能なメッセージングサービスへ、そのフレームワークサービスが待ち受ける P to M メッセージのサブジェクト ID を通知する。それにより、そのフレームワークサービスが待ち受ける P to M サブジェクト ID が、そのメッセージングサービスの管理テーブルに登録される。その後、そのメッセージングサービスは、その管理テーブルに登録されたフレームワークサービスの P to M サブジェクト ID を、そのメッセージングサービスが待ち受ける P to M サブジェクト ID として、他の全てのメッセージングサービスに通知する。これにより、他の全てのメッセージングサービスの管理テーブルにも、そのフレームワークサービスが待ち受ける P to M サブジェクト ID が登録されることになる。結果として、そのフレームワークサービスは、P to M サブジェクト ID をもったメッセージを、いずれのメッセージングサービスからも受信できるようになる。

次に図 9 は、P to P 通信処理を示すフローチャートである。

上述した P to M 通信のサブジェクト ID の管理テーブルへの登録と同様に、各メッセージングサービスの管理テーブルには、そのメッセージングサービスが直接連絡可能なクライアント及びフレームワークサービスが

それぞれ待ち受ける P to P サブジェクト ID、並びに、他の全てのメッセージングサービスがそれぞれ待ち受ける P to P サブジェクト ID が予め自動的に登録されている。

そのような状態において、図 9 に示すように、まず或るクライアント P C 1 は、所定のメッセージングサービス Mes1 とコネクションを確立する。

続いて、クライアント P C 1 から或るサブジェクト ID “A” を持った P to P のリクエストメッセージが送信されると、これを受信したメッセージングサービス Mes1 は、自己の管理テーブル 18 を参照して、そのサブジェクト ID “A” を待ち受けるコンピュータシステムの中の一つ、例えば、フレームワークサービス F r m 1 を選び、それに当該リクエストメッセージを引き渡す（矢印①）。フレームワークサービス F r m 1 は、引き受けたリクエストメッセージのサブジェクト ID “A” に対応したビジネスロジックを呼び出して、そのメッセージの処理を実行し、その実行結果を示すリプライメッセージをメッセージングサービス Mes1 に戻す（矢印③）。そして、メッセージングサービス Mes1 は、そのリプライメッセージをクライアント P C 1 に返信する。そのリプライメッセージを取得したクライアント P C 1 は、メッセージングサービス Mes1 とのコネクションを開放する。

或いは、メッセージングサービス Mes1 は、クライアント P C 1 から受信したサブジェクト ID “A” のリクエストメッセージを、上記の例のようにフレームワークサービス F r m 1 へ送る代わりに、サブジェクト ID “A” を待ち受ける他のメッセージングサービス Mes2 へ送る（矢印②）こともできる。例えば、フレームワークサービス F r m 1 が処理中であるが、メッセージングサービス Mes2 は処理中でないときには、そのようなことが行われる。その場合、メッセージングサービス Mes2 は、そのリクエスト



メッセージを受けると、自己の管理テーブル 18 を参照して、サブジェクト ID “A” を待ち受ける一つのコンピュータシステム、例えば、フレームワークシステム  $Frm2$  を選んで、それに当該リクエストメッセージを送って処理させる（矢印④）。フレームワークシステム  $Frm2$  から処理結果のリプライメッセージが返されると（矢印⑤）、メッセージングサービス  $Mes2$  は、そのリプライメッセージをメッセージングサービス  $Mes1$  に返し（矢印⑥）、そして、メッセージングサービス  $Mes1$  は、そのリプライメッセージをクライアント  $PC1$  へ返す。

上述したように、メッセージングサービス 15 は、自己と直接連絡可能なフレームワークサービス 16 へメッセージを送るだけでなく、他のメッセージングサービスを経由して他のフレームワークサービスへメッセージを送ることも可能である。これにより、複数の  $RtFA$  サーバ 14 間のロードバランシング（負荷平衡）が可能である。

図 10 は、メッセージングサービスがもつロードバランシング機能の説明図である。

図 10 に示すように、各  $RtFA$  サーバ 14 において、各フレームワークサービス 16 は、メッセージングサービス 15 と連絡する複数のスレッド 19 を並列に処理している。また、各メッセージングサービス 15 は、クライアントから受信したメッセージを一時的に保持するメッセージキュー 15a を備えている。

各メッセージングサービス 15 のキュー 15a に蓄積された各メッセージは、そのメッセージングサービス 15 に関連付けられた管理テーブルに従って、そのメッセージのサブジェクト ID に対応する何れかのフレームワークサービス 16 に引き渡される。その場合、メッセージングサービス 15 は、自己に直接連絡可能で且つそのサブジェクト ID を待ち受ける

特定のフレームワークサービス 16 に、そのメッセージを引き渡すことができる。しかし、その時、その特定のフレームワークサービス 16 のスレッド 19 に空きが無い(つまり、処理中である)場合、メッセージングサービス 15 は、その処理中のフレームワークサービス 16 ではなく、そのサブジェクト ID を待ち受ける他のメッセージングサービス 15 に、そのメッセージを引渡す。そのメッセージを受け取った当該他のメッセージングサービス 15 は、当該他のメッセージングサービス 16 が直接に連絡可能であり且つそのサブジェクト ID を待ち受ける特定のフレームワークサービス 16 に、そのメッセージを引き渡す。このようなメッセージ中継を可能にするために、各メッセージングサービス 15 は、自分のメッセージキュー 15 a を常時監視し、そのメッセージキュー 15 a に処理待ちのメッセージが溜まっている場合は、直ちにそのキューからメッセージを取り出し、自分の管理テーブルを参照してそのメッセージの送り先をてきばきと決定して、そのメッセージを送り出す。これにより、複数の RtFA サーバのロードバランスが適正化される。結果として、どのメッセージも可能な限り早期に処理され、よって、従来は夜間にバッチ処理で行われていたバックオフィスの仕事が、リアルタイムに近い態様で早期に実行できる。

続いて、フレームワークサービス 16 の構成について説明する。

図 11 は、フレームワークサービス 16 の構成を示す。

フレームワークサービス 16 の実体は、複数のフレームワークスレッド 25 による並列処理で構成されている。各フレームワークスレッド 25 は、予め準備された一群のビジネスロジック 20 にアクセスし、そして、メッセージングサービス 15 から引き渡されたメッセージのサブジェクト ID に対応する特定のビジネスロジック 22 をロードして実行する。ここで、ビジネスロジックとは、例えば JAVA コードでプログラムされたプロセス

であって、それぞれ固有のメッセージ処理を実行するものである。主要なビジネスロジックの多くは、データベースサーバ 17 に管理されるデータベース 21 に対し、メッセージの内容に応じた所定の操作を要求する。

図 12 は、フレームワークサービス 16 の動作説明図である。

フレームワークサービス 16 のビジネスロジックの動作方式には、大きく分けて同期方式と、非同期方式の 2 種類がある。フレームワークサービス 16 は、クライアントからの或るリクエストメッセージに応答して、ただ 1 つのビジネスロジックのみを実行する場合がある。また、フレームワークサービス 16 は、リクエストメッセージに直接的に응答して 1 つのビジネスロジックを実行し、それに連係して、後続の 1 以上のビジネスロジックを実行する場合もある。図 12 には、クライアント 11, 13 からの仕入登録リクエストのメッセージに直接的に응答して仕入処理のビジネスロジックが実行され、これに連係して、在庫処理及び会計処理という 2 つの後続のビジネスロジックが実行される例が示されている。

クライアントからのリクエストメッセージに直接的に응答してただ 1 つのビジネスロジックのみを実行する場合、フレームワークサービス 16 は、その唯一のビジネスロジックを通常は同期方式で実行し、そして、その実行結果を示すリプライメッセージをクライアントへ返す。

一方、クライアントからのリクエストメッセージに直接的に응答して 1 つのビジネスロジックを実行した後、これに連係して、後続の 1 以上のビジネスロジックを実行する場合、フレームワークサービス 16 は、最初の 1 つのビジネスロジックは通常は同期方式で実行するが、2 番目以降のビジネスロジックは同期方式で実行することもできるし、非同期方式で実行することもできる。図 12 には、最初の仕入処理のビジネスロジックのみが同期方式で実行され、後続の在庫処理と会計処理のビジネスロジックは

非同期方式で実行される例が示されている。図 12 の例のように複数のビジネスロジックが実行される場合、フレームワークサービス 16 は、同期方式で実行したビジネスロジックの全てが完了すると、その実行結果を示すリプライメッセージをクライアントへ返す。非同期方式で実行されるべきビジネスロジックは、クライアントからは切り離されたバックグラウンドで（つまり、クライアントからはオフラインで）実行される。

図 12 に示すように、フレームワークサービス 16 は、フロー定義ファイル 23 を有しており、そこには、そのフレームワークサービス 16 によるビジネスロジックの実行スケジュール（最初に実行すべき 1 つのビジネスロジック名、連係する後続の 1 又はそれ以上のビジネスロジックがある場合にはその後続のビジネスロジックへ渡すメッセージのサブジェクト ID 及び連係方式（同期か非同期か）、など）が、様々なメッセージのサブジェクト ID ごとに定義されている。フレームワークサービス 16 は、一つのメッセージを受け取ると、フロー定義ファイル 23 を参照して、その受信されたメッセージのサブジェクト ID に対応するビジネスロジック実行スケジュールを選択し、選択されたスケジュールに従って 1 以上のビジネスロジックを実行する。その選択されたスケジュールには、少なくとも、そのメッセージに応答して直接的に実行されるべき 1 つのビジネスロジックが記述されているから、フレームワークサービスは、まず、その 1 つのビジネスロジックをロードして実行する。もし、その選択されたスケジュールに、更に、連係する後続の 1 以上のビジネスと連係方式とが記述されていたなら、フレームワークサービスは、最初のビジネスロジックの実行後に、その後続のビジネスロジックを、その連係方式で（つまり、同期的に又は非同期的に）実行することになる。なお、先行のビジネスロジックと後続のビジネスロジックは、同一のフレームワークサービスによって

実行される場合もあるが、上述したロードバランシングにより別のフレームワークサービスによって実行される場合もある。

図 13 は、フロー定義ファイル 23 の簡単な例を示す。

図 13 に示すように、フロー定義ファイル 23 には、サブジェクト ID ごとにビジネスロジック実行スケジュールをそれぞれ記述した複数のセンテンス 231、232、233、234 が存在する。フロー定義ファイル 23 は、例えばテキストファイルであり、よって、定義センテンス 231、232、233、234 の追加、削除、変更などの編集は、テキストエディタなどを用いて簡単に行える。

図 13 の一番上の定義センテンス 231 内に説明されているように、各定義センテンスは、セミコロン “;” で区切られた複数のサブセンテンスから構成される。最初のサブセンテンスには、入力サブジェクト ID、ログキュー名、実行ビジネスロジック名などが含まれる。2 番目以降の各サブセンテンスには、メッセージ変換ビジネスロジック名、出力サブジェクト ID、連携モードなどが含まれる。最初のサブセンテンスは、メッセージに直接的に回答して実行されるべき 1 つのビジネスロジックに関するものであり、これは全ての定義センテンスに必ず記述される。2 番目以降のサブセンテンスは、最初のビジネスロジックに連携して実行される後続の 1 以上のビジネスロジックに関するものであり、よって、後続のビジネスロジックがある場合にのみ記述される。

最初のサブセンテンスの項目の意味は次のとおりである。

ここで、「入力サブジェクト ID」は、フレームワークシステムが受信したメッセージのサブジェクト ID である。

「ログキュー名」は、その受信したメッセージをログに書く場合に用いられるログキューの名称である。例えば、“@DEF” はデフォルトのログキ

ユーを用いることを意味し、“@NONE”はログに書かれないことを意味する。

「実行ビジネスロジック名」は、そのメッセージに直接応答して実行されるべきビジネスロジックの名称である。図示の定義センテンス 2 3 2 ~ 2 3 4 の例では、入力サブジェクト ID と同じコードが用いられている。

2 番目以降の各サブセンテンスの項目の意味は次のとおりである。

「メッセージ変換ビジネスロジック名」は、先行のビジネスロジックの処理結果データを後続のビジネスロジックに渡されるメッセージに変換するために実行されるメッセージ変換ビジネスロジックの名称である。メッセージ変換が不要な場合、ここには“@NONE”と書かれる。

「出力サブジェクト ID」は、後続のビジネスロジックに渡されるメッセージのサブジェクト ID である。

「連係モード」は、後続のビジネスロジックを同期方式で実行するか非同期方式で実行するかを示す。例えば、“S Y N C”は同期方式を意味し、“A S Y N C”は非同期方式を意味する。

図 1 3 の 2 番目の定義センテンス 2 3 2 は次のスケジュール (1) を例示している。

(1) “BuySelBuy”というサブジェクト ID のメッセージ(例えば、仕入検索リクエスト)に回答して、“BuySelBuy”という名称のビジネスロジック(例えば、仕入検索処理)を実行する。

3 番目の定義センテンス 2 3 3 は、次のスケジュール (1) ~ (3) を例示している。

(1) “BuyUpdBuy”というサブジェクト ID のメッセージ(例えば、仕入更新リクエスト)に回答して、まず、“BuyUpdBuy”という名称のビジネスロジック(例えば、仕入更新処理)を実行する。

(2) 次に、“CnvBuyUpdBuy”という名称の変換ビジネスロジックを用い

て、先行する“BuyUpdBuy”ビジネスロジックの出力データのフォーマットを変換して、“InvUpdInv”というサブジェクトIDをもったメッセージ(例えば、在庫更新リクエスト)を作成してメッセージサービスへ出力する。

(3) 出力された“InvUpdInv”というメッセージを処理することになる後続のビジネスロジック(例えば、在庫更新処理)は、同期的に実行される。

さらに、4番目の定義センテンス234は、次のスケジュール(1)~(5)を例示している。

(1) “BuyInsBuy”というサブジェクトIDのメッセージ(例えば、仕入検品更新リクエスト)に応答して、まず、“BuyInsBuy”という名称のビジネスロジック(例えば、仕入検品更新処理)を実行する。

(2) 次に、“CnvBuyInsBuy1”という名称の変換ビジネスロジックを用いて、先行する“BuyInsBuy”ビジネスロジックの出力データのフォーマットを変換して、“InvUpdInv”というサブジェクトIDをもったメッセージ(例えば、在庫更新リクエスト)を作成してメッセージサービスへ出力する。

(3) 出力された“InvUpdInv”というメッセージを処理することになる後続のビジネスロジック(例えば、在庫更新処理)は、非同期的に実行される。

(4) また、“CnvBuyInsBuy2”という名称の変換ビジネスロジックを用いて、先行する“BuyInsBuy”ビジネスロジックの出力データのフォーマットを変換して、“AccDbtBuy”というサブジェクトIDをもったメッセージ(例えば、買掛金登録リクエスト)を作成してメッセージサービスへ出力する。

(5) 出力された“AccDbtBuy”というメッセージを受け取る後続のビジネスロジック(例えば、買掛金登録処理)は、非同期的に実行される。

再び図12を参照する。各フレームワークサービス16は、それに関連

付けられたフロー定義ファイル 23 を参照することにより、処理対象のメッセージに対応するビジネスロジック 22 を選択し実行することができ、また、後続処理の必要性を判断し、必要であれば、後続処理のためのリクエストメッセージを作成してメッセージングサービス 15 へ戻すことができる。

ここで、一のフレームワークスレッドによる処理の後続処理として、他のフレームワークスレッドによる処理を非同期的に行わせる場合、フレームワークサービス 16 は、先行のフレームワークスレッドでのビジネスロジックによる処理の結果（後続処理のためのリクエストメッセージ）を一度メッセージングサービス 15 のキュー 15a にインプットし、その後、他のフレームワークスレッドがメッセージングサービス 15 のキュー 15a からそのリクエストメッセージをゲットすることによって、後続処理を行う。ここで、先行の処理を行うフレームワークシステム 16 と、後続の処理を行なうフレームワークシステム 16 は、同じ場合もあるし、異なる場合もある。各種の処理をどのフレームワークシステム 16 に割り当てるかということは、既に説明したメッセージングサービスのロードバランシング機能によって制御される。

各フレームワークサービス 16 は、メッセージドリブンの構成を探っている。即ち、メッセージングサービス 15 からのメッセージの引渡しをトリガーとしてビジネスロジックのロード及びビジネスロジックによる処理を実行するようになっている。

以上説明した本実施形態によると、フレームワークサービスのビジネスロジックの連係方法に同期方式と非同期方式を用意し、随時に簡単に書き換え可能なフロー定義ファイルの記述によってビジネスロジックの実行スケジュールを制御できるので、複雑なビジネスロジックフローを定義し、



また必要に応じて変更することが容易である。

また、上記実施形態では、メッセージングサービスをメッシュ接続すると共に、キープアライブ機能によって、メッセージングサービス相互間、メッセージングサービスとクライアントとの間、及びメッセージングサービスとフレームワークサービスとの間で、相互にステータス（待ち受けるメッセージのサブジェクト ID、処理中か否か、など）を通知し合う。そして、各メッセージングサービスは、周囲のクライアント、フレームワークサービス及び他のメッセージングサービスの現在のステータスを、管理テーブルに記憶している。それにより、いずれの RtFA サーバがダウンした場合でも、各メッセージングサービスは、正常に使えるメッセージの中継経路を自動的に選択し、メッセージを正しいシステムへ伝えることができる。結果として、フォルトトレラント性が向上し、システムの稼働の信頼性が向上し、システムの拡張が容易になり、更に、システムメンテナンスも容易に行うことができる。

また、メッセージングサービスは、P to P 通信と PtoM 通信のような複数種類のデータ配信方法を備え、管理テーブルに登録されているステータスに応じて、どのメッセージをどのコンピュータシステムにどのデータ配信方法で送るかというメッセージングスケジュールを自動的に選択することができる。その結果、柔軟なメッセージングサービスを提供することができる。

また、メッセージングサービスのロードバランシング機能により、複数の RtFA サーバの負荷バランスを自動的に調整することができ、高いパフォーマンスを提供することができる。

ここで、本発明の範囲は、本実施形態に限られない。例えば、JAVA ベース以外のプラットフォームを用いることによっても同等のシステムの実

現は可能である。また、メッセージングサービスの上述した特徴と、フレームワークサービスの上述した特徴とは、上述した実施形態のように組み合わされている必要は、必ずしも無い。

以下、上記実施形態に具備される幾つかの詳細な機能を説明する。

図 1 4 は、フロー定義ファイル 2 3 の更新システムを示すブロック図である。

フレームワークサービス 1 6 は、定義ファイル 2 3 を更新するためのプログラムモジュールである定義ファイル更新モジュール 1 0 1 を備え、この更新モジュール 1 0 1 は、外部からリロードコマンド 1 1 1 を受けけると、フレームワークサービス 1 6 のメインメモリ 1 0 3 内にロードされている現在使用中の定義ファイル 2 3 を、そこに矢印 1 1 3 で示すように外部記憶装置（例えば、ディスクストレージ 1 0 7 に記憶されている新しい定義ファイル 1 0 9 を上書きすることで、更新する。ディスクストレージ 1 0 7 内の新しい定義ファイル 1 0 9 は、テキストエディタ 1 0 5 などを用いて容易に編集することができる。所望の時に、上述したリロードコマンド 1 1 1 を入力することで、メインメモリ 1 0 3 内の定義ファイル 2 3 が、編集された新しい定義ファイル 1 0 9 と同じ内容に更新される。これにより、フレームワークサービス 1 6 の稼動中であっても所望の時に、定義ファイル 2 3 の内容を書き換えて、これをフレームワークサービス 1 6 の以後の動作に反映させることができる。従って、実行するビジネスロジック 2 2 の種類の変更や、後続の処理を行なうか否か、又は、複数のビジネスロジックの連係した動作を同期的に行うか、非同期的に行うか、といった処理スケジュール変更を、システムを稼動させたままの状態で行うことができる。

図 1 5 は、フレームワークサービスのメッセージ変換処理機能を示す。

フレームワークサービス 16 のためのコンピュータプログラムは例えば JAVA 言語で構築される。図 15 に示すように、フレームワークサービス 16 は、入力メッセージのフォーマット変換を行うための入力変換機能 102 と、出力メッセージのフォーマット変換を行うための出力変換機能 103 とを備えている。入力変換機能 102 は、メッセージングサービス 15 からの例えば XML ファイルの形式の入力メッセージを、JAVA のオブジェクトの形式に変換する。出力変換機能 103 は、メッセージングサービス 15 に出力する JAVA のオブジェクトの形式のメッセージを、XML 形式に変換する。入力変換機能 102 は、XML の DTD (文書型定義) の記述に基づいて JAVA クラスのメンバ変数を取得し、これをインスタンス化することによって JAVA オブジェクト化したメッセージを得る。また、出力変換機能 103 は、JAVA クラスのメンバ変数に基づいて XML の DTD を形成し、XML 化したメッセージを得る。

図 16 は、メッセージングサービスのメッセージキューの構成を示す。

上述の説明では、メッセージキュー 15a を単一のキューであるかのごとくに説明した。しかし、実際には、図 16 に示すように、メッセージキュー 15a には 3 種類のキュー、すなわち、メモリキュー 153 とリングバッファ 154 と DB (データベース) キュー 155 が含まれている。メモリキュー 153 とリングバッファ 154 は、RAM 151 のような高速アクセスが可能な記憶装置に設けられる。DB キュー 155 は、ハードディスク (HDD) のよう不揮発性の大容量の記憶装置に設けられる。

メモリキュー 153 は、メッセージングサービスがクライアントから受信した P to P のリクエストメッセージのように、そのメッセージを処理した結果を受け取る相手が存在し、メッセージの保証が不要であるというメッセージを、一時的に待たせるために使用される。一方、DB キュー 15

5は、メッセージングサービスがフレームワークサービスから受信した非同期的な後続処理のためのリクエストメッセージのように、メッセージの保証が必要なメッセージを、一時的に待たせるために使用される。メッセージの保証が必要か否かは、図2に示したメッセージ10の構成中の「保証モード」で判断される。

リングバッファ154は、P to M通信のメッセージを一時的に待たせるために使用される。各コンピュータシステムは、リングバッファ154を巡回しながらリングバッファ154から自分の待ち受けるメッセージを見つけ出して取っていくことができる。図16では、リングバッファ154は一つのバッファのごとく示されているが、実際には、“H”、“N”及び“L”の3種類の優先順位にそれぞれ対応した3つのリングバッファのセットである。

DBキュー155は、メッセージングサービスがフレームワークサービスから受信した後続処理のためのリクエストメッセージのように、メッセージの保証が必要なメッセージを、一時的に待たせるために使用される。

メッセージングサービスは、上述したキュー153、154、155に対するメッセージの書き込みと読み出しを制御するメッセージキュー管理機能104を有する。メッセージキュー管理機能104は、メモリキュー153をDBキュー155より優先して調べ、メモリキュー153にP to Pメッセージが有れば、そのメッセージをメモリキュー153から取り出して、それを待ち受ける一つのコンピュータシステムへそのメッセージを送る。また、もしメモリキュー153にP to Pメッセージが1つも無ければ、メッセージキュー管理機能104は、DBキュー155を調べ、DBキュー155内にP to Pメッセージが有れば、そのメッセージをDBキュー155から読み、それを待ち受ける一つのコンピュータシステムへそのメ

ッセージを送る。

メッセージキュー管理機能104は、図16に示すように、DBキュー155内のメッセージMに対しては、そのメッセージの処理状態を示すステータス情報Sを付加し、そのメッセージが処理されればステータス情報Sを「未処理」から「処理済」へと書き換える。また、そのメッセージの処理に関してエラーが生じれば、ステータス情報Sを「エラー」とする。エラーの原因としては、データ不整合、プログラムロジックのミスなどがある。DBキュー155に一旦入れられたメッセージは、システムダウンやエラーなどの問題が発生したとしても、消されずにDBキュー155内に保存される。

メッセージキュー管理機能104は、「エラー」のステータス情報Sを「未処理」に書き換える機能を有する。そして、システムダウンやエラーなどの問題が発生した場合、後に、メッセージキュー管理機能104は、ステータス情報Sが「未処理」のメッセージを再度読み出して再送信する。それにより、メッセージの確実な処理が保証される。また、エラーになったメッセージのステータス情報Sを追うことにより、エラーの原因究明も可能である。

また、メッセージキューの管理機能104は、外部コマンドに応じて、「エラー」のステータス情報Sを「処理済」に書き換える機能も備えている。例えば、システムの状況によっては、エラーステータスのメッセージを未処理ステータスに変更して再出力することが適切でない場合がある。また、メッセージに生じたエラーの程度によっては、フレームワークサービスに再出力して処理させるよりも、コマンド入力により当該メッセージに修正を加えた方が適切な場合もある。このような場合、「エラー」のステータス情報Sを「処理済」に書き換える機能が利用される。上述し

た「エラー」のステータス情報 S を「未処理」に書き換える機能と、「処理済」に書き換える機能とは、任意に選択することができる。

さらに、メッセージキュー管理機能 104 は、管理テーブル 18 (図 4) に登録されている全てのコンピュータシステムの各々ごとに、リングバッファ 155 の読みしアドレスポインタを有し、各コンピュータシステム用の読み出しアドレスポインタでリングバッファ 155 中を巡回しながら、そのコンピュータシステムが待ち受ける P to M メッセージをリングバッファ 155 から見つけ出し、その見つかった P to M メッセージを読んで、そのコンピュータシステムへ送る。

図 17 及び図 18 は、メッセージキュー管理機能 104 がメッセージキューにメッセージを格納する動作フローを示す。

図 17 に示すように、メッセージキュー管理機能 104 は、クライアントからメッセージを受信すると (161)、まず、そのメッセージに関して、クライアントから非トランザクションモード (NON-TRAN モード) とトランザクションモード (TRAN モード) の何れが指定されているかを判断する (163)。ここで、NON-TRAN モードが指定されている場合、メッセージキュー管理機能 104 は、メッセージを受信すれば直ちに、そのメッセージをメッセージキューに格納する (つまり、そのメッセージの処理を進める)。一方、TRAN モードが指定されている場合、メッセージキュー管理機能 104 は、メッセージを受信しても、直ちにはそのメッセージをメッセージキューには格納せず、別の一時キューに溜めておき、クライアントから COMMIT (実行) コマンドが送られると、そのときにはじめて、そのメッセージをメッセージキューに格納する (つまり、そのメッセージの処理を進める)。クライアントは、TRAN モードを指定してリクエストメッセージを発した場合、COMMIT コマンドを発しない限り、そのリクエストメッセー

ジを後で撤回したり修正したりすることが可能である。

図 17 のステップ 163 の判断の結果が NON-TRAN モードである場合、メッセージキュー管理機能 104 は、その受信したメッセージが P to P 通信のものか P to M 通信のものかを判断する (165)。その判断の結果が P to P 通信である場合、メッセージキュー管理機能 104 は、その受信したメッセージをキューに入れる (167)。キューに入れられたメッセージは、すぐに読み出されてフレームワークサービスへ送られてそこで処理される。その処理の結果として、フレームワークサービスから後続処理のリクエストメッセージが返された場合、メッセージキュー管理機能 104 は、その後続処理のリクエストメッセージを DB キューに入れる (169)。そして、メッセージキュー管理機能 104 は、その受信メッセージの処理結果を示す成功メッセージをクライアントへ返す (171)。

ステップ 165 の判断の結果が P to M 通信である場合、メッセージキュー管理機能 104 は、その受信したメッセージの優先順位(図 2)が“H”、“N”又は“L”の何れであるか判別し (173)、そして、判別された優先順位に対応したリングバッファに、その受信メッセージを格納する (175、177 又は 179)。リングバッファに入れられた P to M メッセージは、それを待ち受ける全てコンピュータシステムに送信されることになる。そして、メッセージキュー管理機能 104 は、その受信メッセージについて成功メッセージをクライアントへ返す (181)。

ステップ 163 の判断結果が NON-TRAN モードであった場合、処理は図 8 に示すステップ 181 へ進む。ステップ 181 で、メッセージキュー管理機能 104 は、その受信メッセージを上述べたメッセージキューとは別の一時キューに格納する。その後、クライアントから、その受信メッセージに関して COMMIT コマンドを受信すると、メッセージキュー管理機能 1

04は、一時キューからそのメッセージを取り出して、そのメッセージについてステップ185以降の処理を行う。ステップ185以降の処理は、既に説明した図7中のステップ165以降の処理と同じである。

図19は、メッセージキュー管理機能104のメッセージ読出機能を示す。

図19に示すように、メッセージングサービス15のメッセージキュー管理機能104は、クライアント及びフレームワークサービス16からそれぞれ受信したメッセージを、上述したような構成をもつメッセージキュー15aに蓄積する。メッセージキュー管理機能104は、メッセージキュー15aのうち特に図16に示したメモリキュー153及びDBキュー155からメッセージを取り出す動作に関して、優先モード104aとシーケンス保証モード104bの2種類の機能を有している。優先モード104aは、メッセージの優先順位に基づいてメッセージキューからのメッセージの取り出し順序を制御する。一方、シーケンス保証モード104bは、メッセージキューから先に取り出されたメッセージの処理がフレームワークサービスにおいて完了するまで、当該メッセージキューに蓄積されている後続のメッセージの取り出しを禁止することにより、複数メッセージの一定の処理順序を保証する。各メッセージのサブジェクトID)に応じて、又は予め設定された各メッセージキューのモード設定に応じて、各メッセージキューについて優先モード104aを実行するか、シーケンス保証モード104bを実行するか、が選択される。シーケンス保証モード104bは、例えば、複数のクライアントからそれぞれ売上処理のリクエストメッセージが来たとき、先にリクエストされた売上処理が完了した後に後から来た売上処理のリクエストを実行するというような一定の処理順序を保証したい場合に、適用される。



図 20 は、優先モード 104a のための記憶域の構成を示す。図 21 は、優先モード 104a の動作の流れを示す。

図 20 に示すように、優先モード 104a は、ソータブルスタック 211 とシンプルスタック 213 という 2 つのスタックメモリと、“H”、“N” 及び “L” という 3 つの優先順位にそれぞれ対応した例えば 3 つのエントリ 215H、215N 及び 215L とを使用する。なお、エントリの数をもっと多くすることもできる。例えば、図示の 3 つのエントリの他に、“H” に対応したエントリをもう一つ追加するというようにである。

初期的に、各エントリ 215H、215N 及び 215L には、それぞれ、インデックスとして、例えば図示のような “5”、“3” 及び “1” という数値が設定される。インデックスの初期値が大きいくほど、優先的レベルが高いことを意味する。

そして、初期的に、全てのエントリ 215H、215N 及び 215L が、図示のようにソータブルスタック 211 に積まれ、一方、シンプルスタック 213 は空である。

ソータブルスタック 211 は、そこに積まれている複数のインデックスの配列を、インデックスの大きい順に、最も大きいインデックスをもったエントリが一番上に、最も小さいインデックスをもったエントリが一番下になるように、自動的に並べ替える。従って、ソータブルスタック 211 からは、インデックスのより大きいエントリほど、より先に取り出される。

シンプルスタック 213 は、単純に、後から入れられたエントリを先に入れられたエントリの上に積む。従って、シンプルスタック 213 からは、入れられた時がより後のエントリほど、より先に取り出される。

優先モード 104a は、図 21 に示すフローに従って、ソータブルスタック 211 とシンプルスタック 213 との間でのエントリの移動と、エン

トリのインデックスの操作を行なうことにより、メッセージキュー（メモリキュー 1 5 3 又は DB キュー 1 5 5）からメッセージを取り出す順序を制御する。

以下、図 2 1 に示す流れを説明する。まず、優先モード 1 0 4 a は、ソータブルスタック 2 1 1 から、一番上の（つまり、インデックスが最も大きい）1 つのエントリを取り出し（2 2 1）、そのエントリをシンプルスタック 2 1 3 に積む（2 2 3）。次に、優先モード 1 0 4 a は、シンプルスタック 2 1 3 の一番上のエントリの優先順位とマッチするメッセージがメッセージキューに有るか否かをチェックする（2 2 5）。その結果、マッチするメッセージが無ければ、処理はステップ 2 2 1 へ進む。また、上記チェックの結果、マッチするメッセージがあれば、優先モード 1 0 4 a は、そのマッチしたメッセージをメッセージキューから取り出す（2 2 7）。取り出されたメッセージは、何れかのフレームワークシステムにて処理されることになる。その後、優先モード 1 0 4 a は、シンプルスタック 2 1 3 内のエントリ数が 1 つか否かをチェックする（2 2 9）。その結果、シンプルスタック 2 1 3 内のエントリ数が 2 つ以上であれば、シンプルスタック 2 1 3 から全てのエントリを順に取り出してソータブルスタック 2 1 1 に積む（2 3 7）。そして、優先モード 1 0 4 a は、ステップ 2 2 7 でメッセージキューから取り出したメッセージについて、フレームワークシステムからの処理結果を示すリプライメッセージをクライアントへ返す（2 3 9）。

ステップ 2 2 9 でのチェックの結果、シンプルスタック 2 1 3 内のエントリ数が 1 つである場合、優先モード 1 0 4 a は、シンプルスタック 2 1 3 内のエントリのインデックスを 1 だけ減らす（2 3 1）。そして、優先モード 1 0 4 a は、その減らされたインデックスが“−1”であるかどうか

かチェックし（２３３）、“－１”でなければ、処理をステップ２３７へ進め、一方、“－１”であれば、そのインデックスを初期値に戻して（２３５）から、処理をステップ２３７へ進める。

以上の操作により、メッセージキュー内のメッセージの並び順序をそれらのメッセージの優先順位を考慮して修正した順序で、メッセージキューからメッセージの取り出されることになる。すなわち、メッセージキューに入っているメッセージの順序が、優先順位において、例えば図２０に示すような“Ｈ”、“Ｈ”、“Ｌ”、“Ｈ”、“Ｈ”、“Ｈ”、“Ｎ”であった場合、そのメッセージキューから取り出されるメッセージの順序は、優先順位において、“Ｈ”、“Ｈ”、“Ｎ”、“Ｈ”、“Ｈ”、“Ｈ”、“Ｌ”となる。

図２２は、シーケンス保証モード１０４ｂの動作の流れを示す。

図２２に示すように、シーケンス保証モード１０４ｂは、メッセージの取り出し要求が生じると（２４１）、まず、メッセージキューをロックして（２４３、２４５、２５７）、メッセージキューからの他のメッセージ取り出しを禁止する。次に、シーケンス保証モード１０４ｂは、メッセージキューから１つのメッセージを読出す（２４７、２４９）。読み出されたメッセージは、何れかのフレームワークシステムで処理され（２５１）、その処理により然るべきデータベースの更新のコミット（又は処理失敗の場合にはロールバック）が行われる（２５３）。その後、シーケンス保証モード１０４ｂは、メッセージキューのロックを解除する（２５５）。

上記の操作により、先行するメッセージの処理が完了した後に、次のメッセージの処理が実行されるため、メッセージの一定の処理順序が保証される。

図２３は、フレームワークサービス１６のメッセージ判別機能を示す。

図２３に示すように、フレームワークサービス１６は、メッセージ判別

ビジネスロジック 106 を有し、メッセージングサービス 15 からメッセージを受け取ると、メッセージ判別ビジネスロジック 106 を実行する。メッセージ判別ビジネスロジック 106 は、その受信メッセージのサブジェクト ID にマッチするサブジェクト ID が書かれた定義センテンスを、そのフレームワークシステム 16 に関連付けられたフロー定義ファイル 23 から探す。マッチした定義センテンスが見つければ、次に、その定義センテンスに記述されている実行ビジネスロジック 22 が実行されることになる。しかし、マッチした定義センテンスが見つからなければ、メッセージ判別ビジネスロジック 106 は、処理できない旨のエラーのリプライメッセージをメッセージングサービス 15 へ返信し、そのリプライメッセージはクライアントへ送られる。既に説明したように、メッセージングサービス 15 は、管理テーブル 18 (図 4) に基づいて、各メッセージをそのメッセージを待ち受けるコンピュータシステムにのみ送信するから、原則的には、或るメッセージが誤ったフレームワークシステムに送られることはない。しかし、万が一、クライアントからの或るメッセージが誤ったフレームワークサービスに送られることが発生したとしても、クライアントは直ちに処理できない旨のエラーのリプライメッセージを受け取れるので、長々と待たされることなく、速やかに次の行動に移ることができる。また、フレームワークサービス 16 は、予め登録された識別情報を有するメッセージに限ってフレームワークサービスでの処理を許可するサービス停止中モードを備え、当該モードが設定されている場合に予め登録されていない識別情報を有するメッセージをクライアントから受信した場合は、当該メッセージを受け付けられない旨のメッセージをクライアントに返信する。

図 24 は、フレームワークサービスがもつメッセージ変換ビジネスロジ

ックを示す。

図 2 4 に示すように、フレームワークサービス 1 6 は、1 又はそれ以上のメッセージ変換ビジネスロジック 1 0 5 を有している。それらのメッセージ変換ビジネスロジック 1 0 5 は、例えば、図 1 3 に示したフロー定義ファイル 2 3 の定義センテンス 2 3 3 及び 2 3 4 に記述された“CnvBuyUpdBuy”、“CnvBuyInsBuy1”及び“CnvBuyInsBuy2”などである。各メッセージ変換ビジネスロジック 1 0 5 は、予め定められた 1 つのビジネスロジック B L 1 に対応している。或る定義センテンスに従って或るビジネスロジック B L 1 が実行されると、引き続いて、これに対応したメッセージ変換ビジネスロジック 1 0 5 が実行される。メッセージ変換ビジネスロジック 1 0 5 は、先行のビジネスロジック B L 1 から出力される処理結果のメッセージの形式を、後続のビジネスロジック B L 2 への入力メッセージの形式に適合させるフォーマット変換機能と、先行のビジネスロジック B L 1 から出力されたメッセージのパラメータに応じて、このメッセージを後続ビジネスロジック B L 2 に入力するか否かを判定するメッセージフィルタリング機能とを備えている。例えば、先行のビジネスロジック B L 1 が扱うメッセージのパラメータ配置と、後続ビジネスロジック B L 2 が扱うメッセージのパラメータ配置が異なる場合、メッセージ変換ビジネスロジック 1 0 5 は、前者のパラメータ配置を後者のパラメータ配置に変換する。また、先行ビジネスロジック B L 1 の出力メッセージの内容が特別のもの（例えば、リクエストの拒否、データベースの更新失敗、トランザクションのロールバックなど）であり、そのため、後続のビジネスロジック B L 2 の実行が無意味であるという場合がある。そのような場合、メッセージ変換ビジネスロジック 1 0 5 は、メッセージフィルタリング機能により、後続ビジネスモデル B L 2 への入力メッセージの生成をオ

ミットする。ここで、先行ビジネスロジック B L 1 とメッセージ変換ビジネスロジック 1 0 5 との間、又はメッセージ変換ビジネスロジック 1 0 5 と後続のビジネスロジック B L 2 との間には、メッセージを一時保持するメッセージキューを介在させてもよい。

以上の説明は、本発明の説明のための例示にすぎない。本発明は、上述した実施形態のみに限らず、他の多くのシステム構成のバリエーションにおいても実施することができる。

## 請 求 の 範 囲

1. クライアントと通信可能に接続されるフレームワークシステムにおいて：

複数のビジネスロジックと；

前記複数のビジネスロジックと関連付けられ、前記クライアントからのリクエストのメッセージに応答して、選択された1又はそれ以上のビジネスロジックを実行し、そして、前記クライアントへのリプライのメッセージを出力するフレームワークサービスと；

前記クライアントと前記フレームワークサービスとの間に介在し、前記クライアントと前記フレームワークサービスとの間でメッセージを中継するメッセージングサービスと；

前記フレームワークサービスに関連付けられたフロー定義ファイルと  
を備え；

前記リクエストのメッセージは、前記リクエストのメッセージのサブジェクトに関するサブジェクト ID を含んでおり；

前記フロー定義ファイルは、複数の異なるサブジェクト ID にそれぞれ対応した複数の定義センテンスを含んでおり、各定義センテンスは所定の1又はそれ以上のビジネスロジックのための実行スケジュールを記述しており；

前記フレームワークサービスは、前記メッセージングサービスから前記リクエストのメッセージを受け取ると、前記定義ファイル内の前記リクエストのメッセージのサブジェクト ID に対応した定義センテンスを参照し、参照された定義センテンスに記述された実行スケジュールに従って、実行するべき1又はそれ以上のビジネスロジックを選択する；

フレームワークシステム。

2. 請求項1記載のフレームワークシステムにおいて、前記フレームワークサービスが稼動している間に、前記フロー定義ファイルを更新するフロー定義更新コンポーネントを更に備えたフレームワークシステム。

3. 請求項1記載のフレームワークシステムにおいて、

前記フロー定義ファイル内の少なくとも一つの定義センテンスは、一つの先行のビジネスロジックと少なくとも一つの後続のビジネスロジックとを含む複数のビジネスロジックの連係された実行のための連係実行スケジュールを記述しており、前記連係実行スケジュールは、前記後続のビジネスロジックの実行に関して同期方式と非同期方式の選択に関する連係モードを含んでおり；

前記フレームワークサービスは、前記連係実行スケジュールに従って前記複数のビジネスロジックを実行する場合、前記連係実行スケジュール内の前記連係モードに従って、前記先行のビジネスロジックの実行と同期的に又は非同期的に、前記後続のビジネスロジックを実行する；  
フレームワークシステム。

4. 請求項3記載のフレームワークシステムにおいて、

前記連係実行スケジュールは、前記先行のビジネスロジックに関する情報と、前記後続のビジネスロジックのための第2のリクエストのメッセージに関する情報とを含み；

前記フレームワークシステムは、前記連係実行スケジュールに従って前記複数のビジネスロジックを実行する場合、前記先行のビジネスロジックを実行した後、前記先行のビジネスロジックの実行結果を用いて前記第2のリクエストのメッセージを生成して、前記第2のリクエストのメッセー



ジを前記メッセージングサービスに送り、その後、前記メッセージングサービスから前記第 2 のリクエストのメッセージを受けると、前記第 2 のリクエストのメッセージに対応する定義センテンスの実行スケジュールに従って、前記後続のビジネスロジックを実行する；  
フレームワークシステム。

5. 請求項 4 記載のフレームワークシステムにおいて、

前記メッセージングサービスが、前記第 2 のリクエストのメッセージを  
保証するための不揮発性のメッセージキューを備え；

前記メッセージキューは、前記第 2 のリクエストメッセージが前記フレームワークサービスへ送られた後も、前記第 2 のリクエストメッセージを  
消さずに保存する；  
フレームワークシステム。

6. 請求項 5 記載のフレームワークシステムにおいて、

前記メッセージングサービスが、前記メッセージキューで保存されて  
いる前記第 2 のリクエストのメッセージを、必要に応じて、前記フレーム  
ワークシステムへ再送信する；  
フレームワークシステム。

7. クライアントと通信可能に接続されるフレームワークシステムにおいて：

前記クライアントからのリクエストのメッセージを処理し、そして、  
前記クライアントへのリプライのメッセージを出力するフレームワーク  
サービスと；

前記クライアントと前記フレームワークサービスとの間に介在し、  
前記クライアントと前記フレームワークサービスとの間でメッセージを  
中継するメッセージングサービスと

を備え；

前記メッセージングサービスは、前記リクエストのメッセージと前記リブライのメッセージを含む P to P 通信のメッセージだけでなく、前記クライアントと前記フレームワークサービスとの間の P to M 通信のメッセージをも中継する；  
フレームワークシステム。

8. 請求項 7 記載のフレームワークシステムにおいて、

前記メッセージングサービスが、前記 P to M 通信のメッセージを一時的に待たせるためのリングバッファを有する；  
フレームワークシステム。

9. 請求項 7 記載のフレームワークシステムにおいて、

前記 P to P 通信のメッセージの各々には、メッセージ保証の要否に関する保証モードが含まれており；

前記メッセージングサービスが、メッセージ保証が不要な P to P 通信のメッセージを一時的に待たせるための第 1 のメッセージキューと、メッセージ保証が必要な P to P 通信のメッセージを一時的に待たせるための不揮発性の第 2 のメッセージキューとを有する；  
フレームワークシステム。

10. クライアントと通信可能に接続されるフレームワークシステムにおいて、

前記クライアントと接続された少なくとも一つのメッセージングサービスを含んだ、相互にメッシュ接続されている複数のメッセージングサービスと；

前記複数のメッセージングサービスにそれぞれ接続された複数のフレームワークサービスと

を備え；

前記複数のフレームワークサービスの各々は、予め定められたサブジェクト ID をもったリクエストのメッセージを待ち受け、前記予め定められたサブジェクト ID をもったリクエストのメッセージを受信すると、受信されたリクエストのメッセージを処理し、そして、前記クライアントへのリプライのメッセージを出力することができ；

前記複数のメッセージンサービスは、前記クライアントと前記複数のフレームワークサービスの間でのメッセージの中継、及び、前記複数のフレームワークシステム相互間でのメッセージの中継を行う；  
フレームワークシステム。

1 1. 請求項 1 0 記載のフレームワークシステムにおいて、

各メッセージングサービスが、他の 1 又はそれ以上のメッセージングサービスの稼動を監視し、他の或るメッセージングサービスについて正常な稼動を検出できなかった場合は、前記他の或るメッセージングサービスへメッセージを中継する代わりに、他の正常に稼動しているメッセージングサービスへメッセージを中継する；  
フレームワークシステム。

1 2. 請求項 1 0 記載のフレームワークシステムにおいて、

各メッセージングサービスが、前記各メッセージングサービスに接続されたフレームワークサービスの状態を監視し、監視された前記状態に応じて、前記クライアントからの前記リクエストのメッセージを前記各メッセージングサービスに接続されたフレームワークサービスに中継するか、他の何れかのメッセージングサービスに中継するかを選択する；  
フレームワークシステム。

1 3. 請求項 1 0 記載のフレームワークシステムにおいて、

各メッセージングサービスは独自の管理テーブルを有し；

前記各メッセージングサービスの独自の管理テーブルには、前記各メッセージングサービスに接続されたフレームワークシステムが待ち受けるメッセージのサブジェクト ID、及び、前記各メッセージングサービスに接続された他の 1 又はそれ以上のメッセージングサービスがそれぞれ待ち受けるメッセージのサブジェクト ID が登録されており；

前記各メッセージングサービスは、メッセージを受信したとき、前記各メッセージングサービスの独自の管理テーブルを参照して、受信されたメッセージのサブジェクト ID にマッチするサブジェクト ID を待ち受けるフレームワークサービス又は他のメッセージングサービスを探し、見つかったフレームワークサービス又は他のメッセージングサービスへ前記受信されたメッセージを中継する；

フレームワークシステム。

14. 請求項 13 に記載のフレームワークシステムにおいて、

各メッセージングサービスは、前記各メッセージングサービスの独自の管理テーブルの内容に基づいて、前記各メッセージングサービスが待ち受けるメッセージのサブジェクト ID を、前記各メッセージングサービスに接続されている他の 1 又はそれ以上のメッセージングサービスに対して通知する；

フレームワークシステム。

15. 請求項 14 に記載のフレームワークシステムにおいて、

各メッセージングサービスは、前記各メッセージングサービスに接続されている他の各メッセージングサービスの各々から、前記他の各メッセージングサービスが待ち受けるメッセージのサブジェクト ID を通知されたとき、通知された内容が前記各メッセージングサービスの独自の管理

テーブルに未だ登録されていなければ、前記通知された内容を前記独自の管理テーブルに追加登録する；

フレームワークシステム。

16. クライアントと通信可能に接続されるフレームワークシステムにおいて；

前記クライアントからのリクエストのメッセージを処理し、そして、前記クライアントへのリプライのメッセージを出力するフレームワークサービスと；

前記クライアントと前記フレームワークサービスとの間に介在し、前記クライアントと前記フレームワークサービスとの間でメッセージを中継するメッセージングサービスとを備え；

前記リクエストのメッセージは所与の優先順位を有しており；

前記メッセージングサービスが、前記リクエストのメッセージを一時的に待たせるためのメッセージキューと、前記メッセージキューの入出力を管理するキュー管理コンポーネントとを備え；

前記キュー管理コンポーネントは、

前記メッセージキューに複数のメッセージが蓄積されているとき、各メッセージの優先順位に応じて、前記複数のメッセージの前記メッセージキューからの出力順序を制御する優先モードと、

前記メッセージキューに複数のメッセージが蓄積されているとき、前記メッセージキューから先に取り出されたメッセージの処理が前記フレームワークサービスにおいて完了するまで、前記メッセージキューに蓄積されている他のメッセージの取り出しを禁止するシーケンス保証モードと

を備える；

フレームワークシステム。

17. クライアントと通信可能に接続されるフレームワークシステムの動作方法において：

複数の異なるサブジェクト ID にそれぞれ対応したビジネスロジック実行スケジュールが記述されたフロー定義ファイルを用意するステップと；

前記クライアントから、所与のサブジェクト ID をもつリクエストのメッセージを受信するステップと；

前記フロー定義ファイル内の、受信されたリクエストのメッセージのサブジェクト ID に対応したビジネスロジック実行スケジュールを参照するステップと；

参照されたビジネスロジック実行スケジュールに従って、予め用意された複数のビジネスロジックの中から、1 又はそれ以上のビジネスロジックを選択するステップと；

選択された 1 又はそれ以上のビジネスロジックを実行するステップと；

前記クライアントへリプライのメッセージを返すステップと；  
を備えた方法。

18. クライアントと通信可能に接続されるフレームワークシステムの動作方法において：

一つのクライアントと一つのフレームワークサービスとの間で、リクエストとリプライのメッセージをやりとりする P to P 通信のステップと；

同じメッセージを複数のクライアント又は複数のフレームワーク

サービスへ送る P to M 通信のステップと；  
を備えた方法。

19. クライアントと通信可能に接続されるフレームワークシステムの動作方法において：

相互にメッシュ接続されている複数のメッセージングサービスの中の一つが、前記クライアントからリクエストのメッセージを受信するステップと；

前記一つのメッセージングサービスが、受信されリクエストのメッセージを、前記一つのメッセージングサービスに接続されたフレームワークサービス、及び前記一つのメッセージングサービスに接続された他の何れか一つのメッセージングサービスの内の選択された一方へ、中継するステップと；

前記受信されたリクエストのメッセージが前記他の一つのメッセージングサービスへ中継されたとき、前記他の一つのメッセージングサービスが、前記受信されリクエストのメッセージを、前記他の一つのメッセージングサービスに接続されたフレームワークサービス、及び前記他の一つのメッセージングサービスに接続された更に他の何れか一つのメッセージングサービスの内の選択された一方へ、中継するステップと；  
を備えた方法。

20. クライアントと通信可能に接続されるフレームワークシステムの動作方法において：

前記クライアントからリクエストのメッセージを受信するステップと；

受信されたリクエストのメッセージをメッセージキューに入れるステップと；

前記メッセージキューからリクエストのメッセージを取り出すステップと；

取り出されたリクエストのメッセージを処理するステップとを備え；

前記取り出すステップは、

前記メッセージキューに複数のメッセージが蓄積されているとき、各メッセージの優先順位に応じて、前記複数のメッセージの前記メッセージキューからの出力順序を制御する優先モードと、

前記メッセージキューに複数のメッセージが蓄積されているとき、前記メッセージキューから先に取り出されたメッセージの処理が完了するまで、前記メッセージキューに蓄積されている他のメッセージの取り出しを禁止するシーケンス保証モードとを有した方法。



FIG. 1

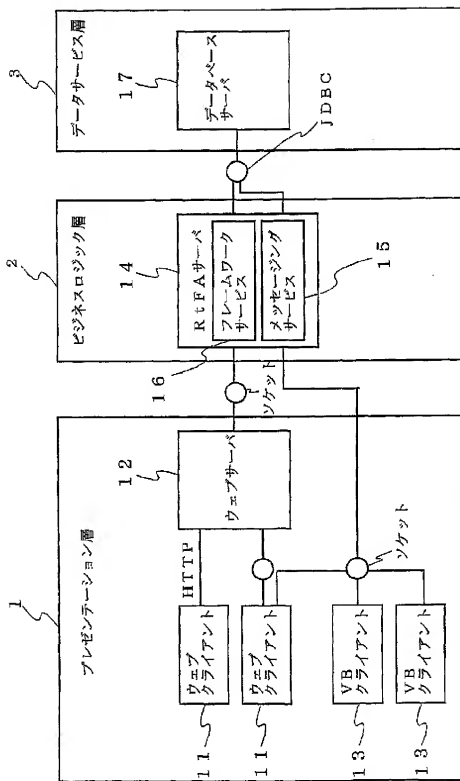


FIG. 2

10

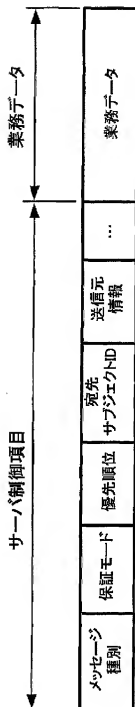


FIG. 3

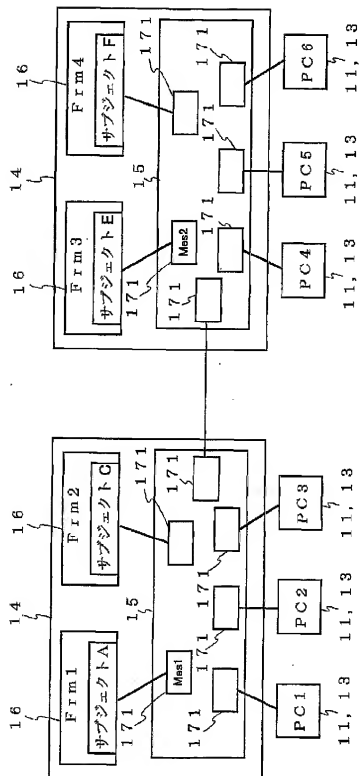


FIG. 4

18

コンピュータ 種別	コンピュータ ID	IPアドレス	PtoP サブジェクトID	PtoM サブジェクトID	処理中 フラグ
F	Frm1	10.100.1.1	A,B,C,D	O,P,Q,R	1
F	Frm2	10.100.1.2	C,D,E,F	P,Q,R,S	0
M	Mes2	10.100.2	D,E,F,G,H,I,J	P,Q,R,S,T,U	0
T	PC1	10.1.1.3	K,L	V,W	0
T	PC2	10.1.1.2	K,L	V,W	1
T	PC3	10.1.1.3	M,N	X,Y	0

FIG. 5

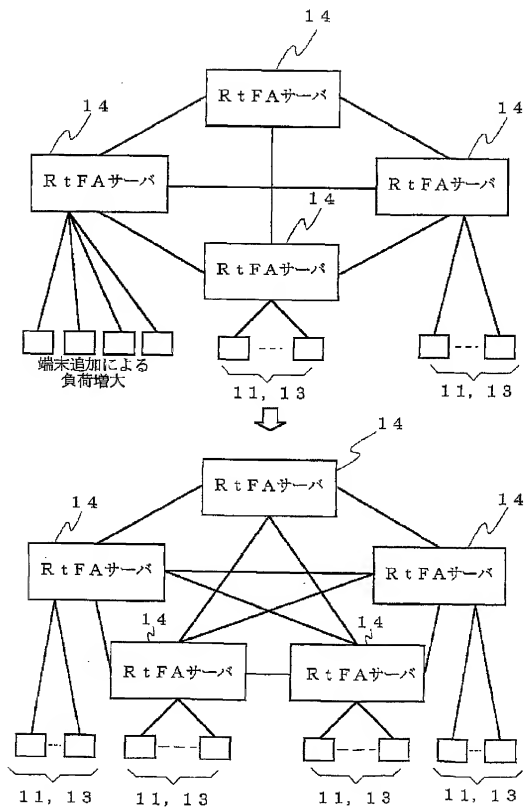


FIG. 6

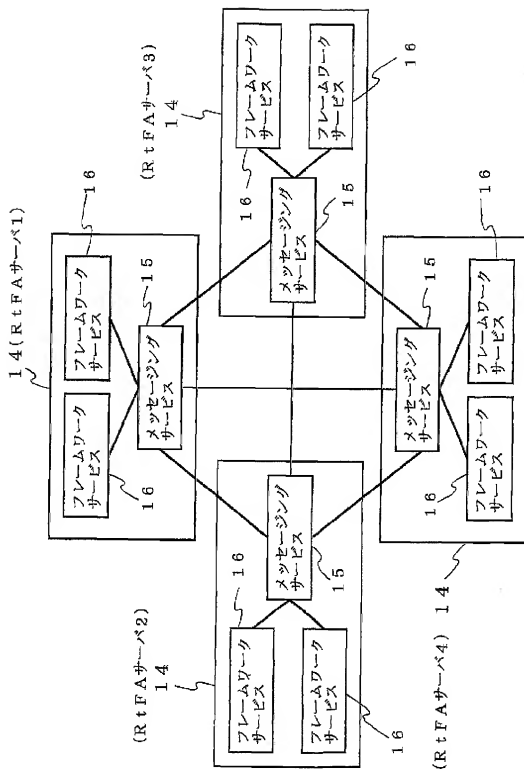


FIG. 7

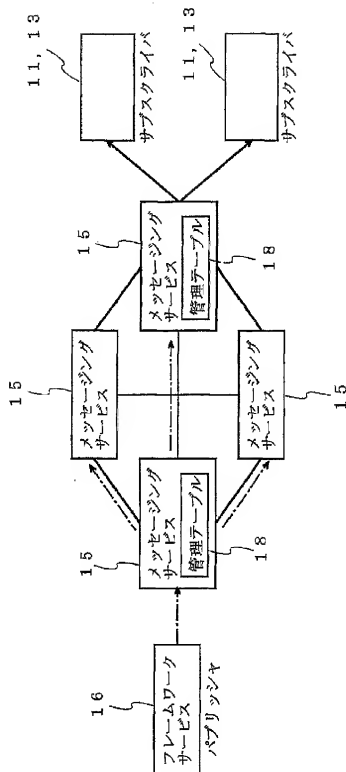






FIG. 9

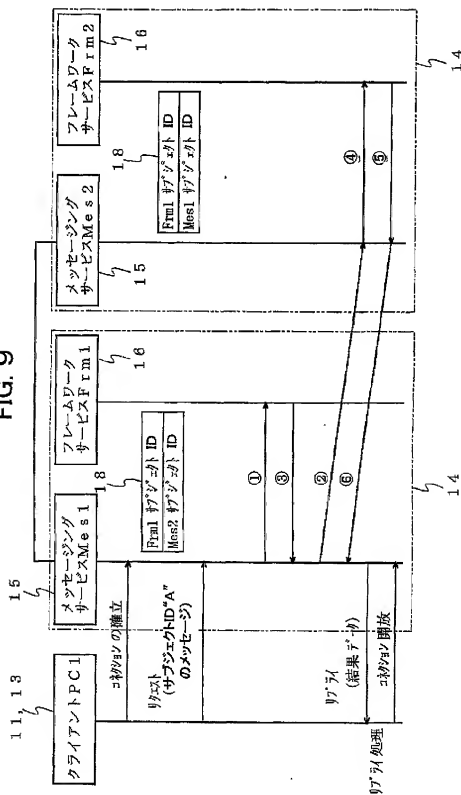


FIG. 10

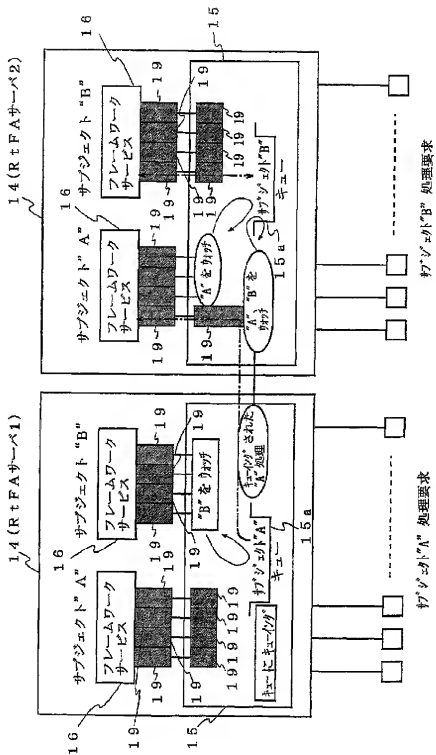


FIG. 11

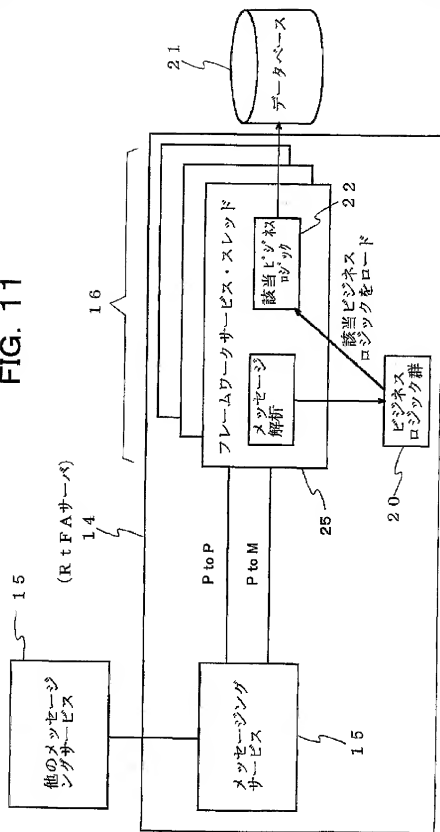


FIG. 12

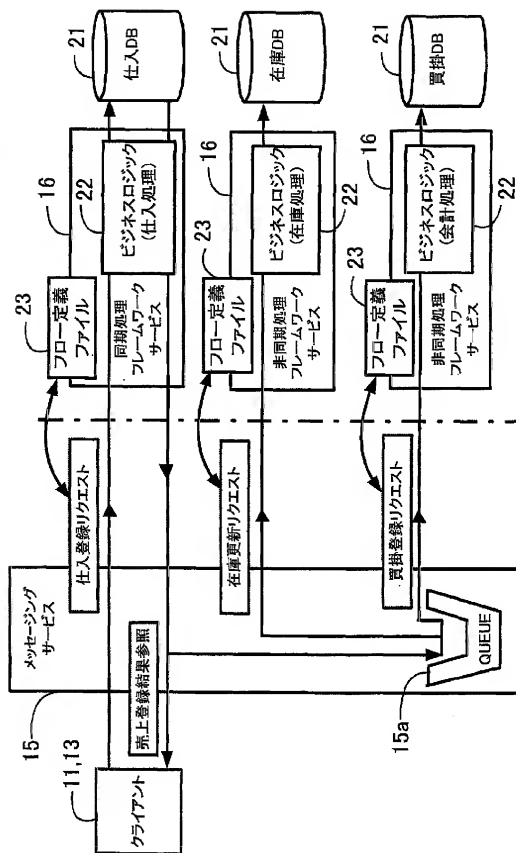


FIG. 13

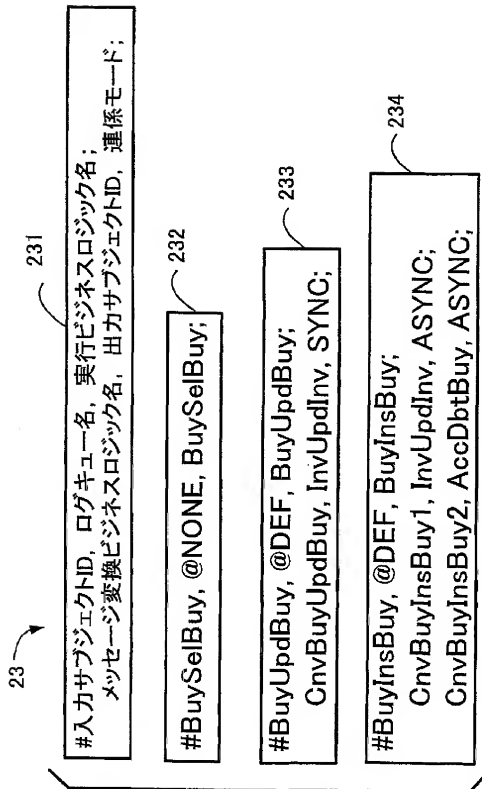


FIG. 14

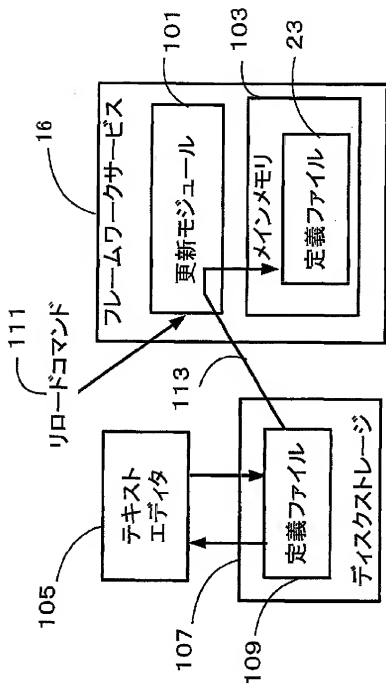
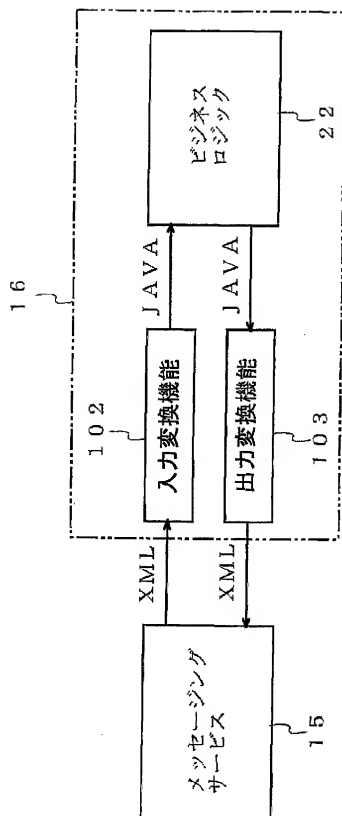


FIG. 15



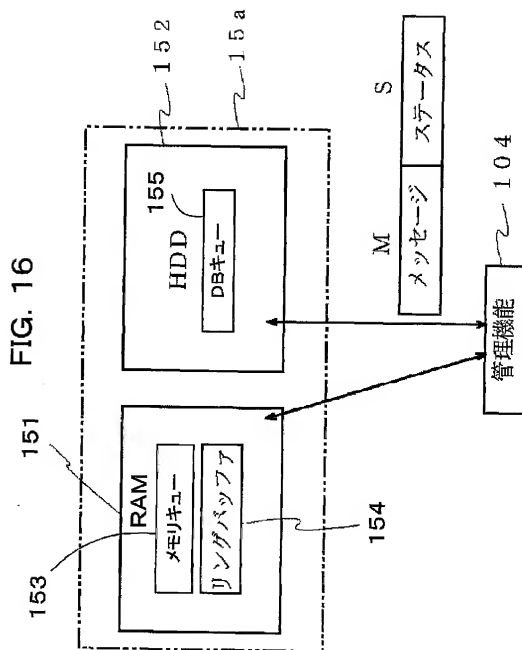




FIG. 17

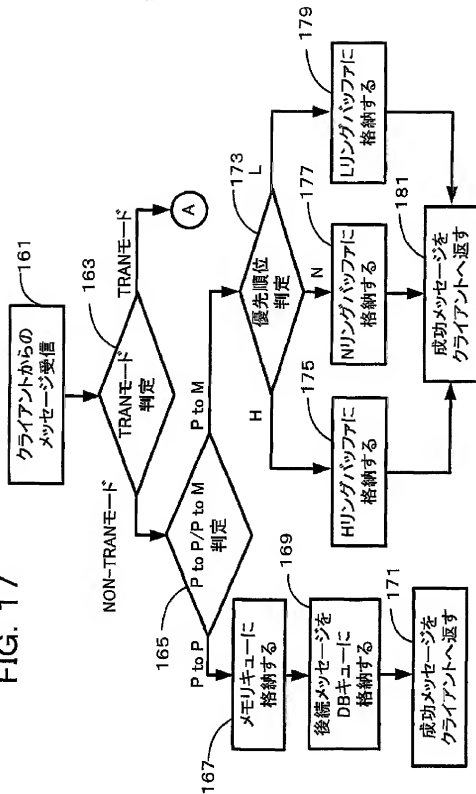


FIG. 18

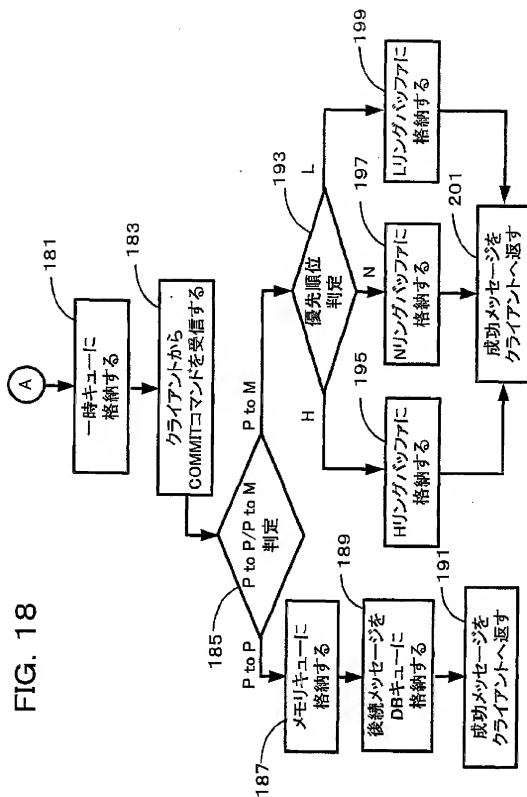


FIG. 19

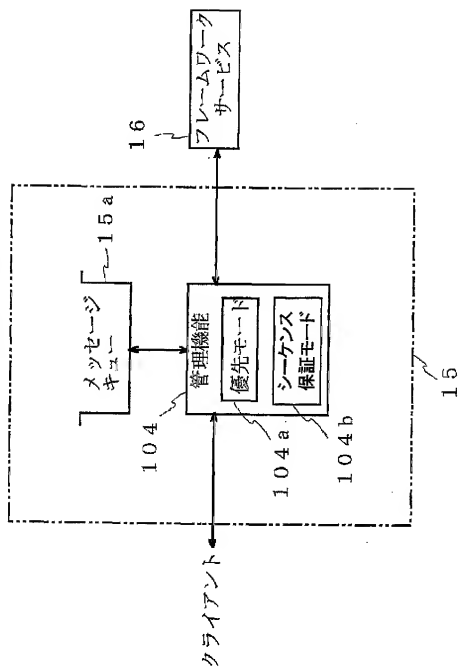


FIG. 20

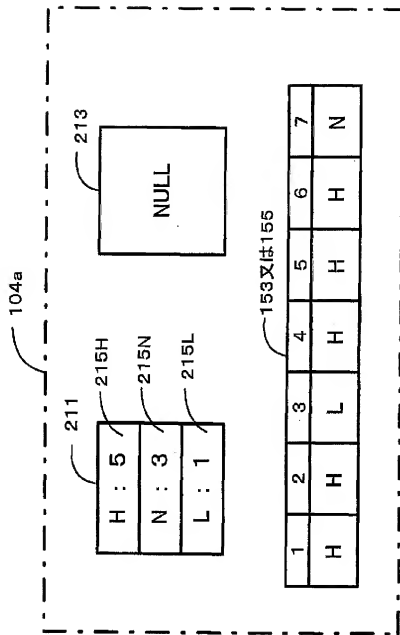


FIG. 21

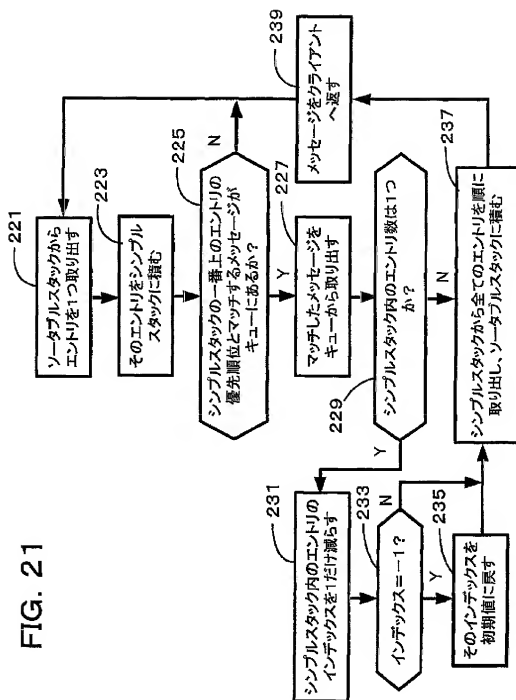


FIG. 22

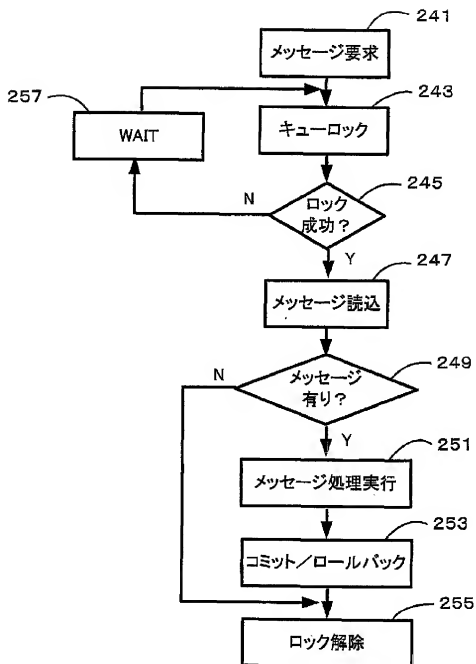


FIG. 23

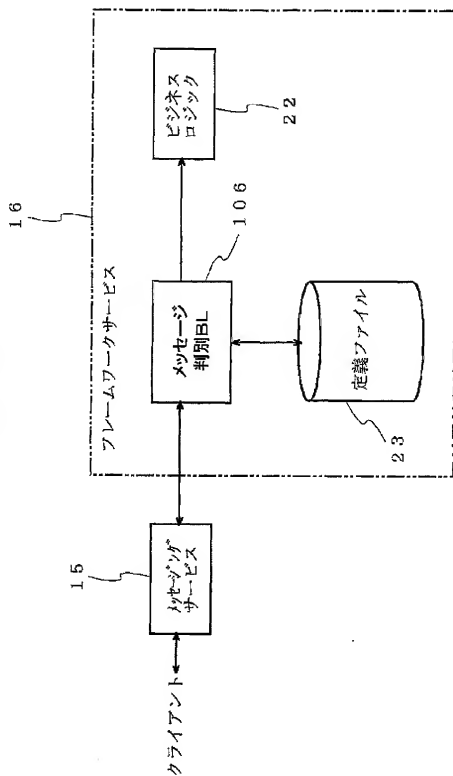
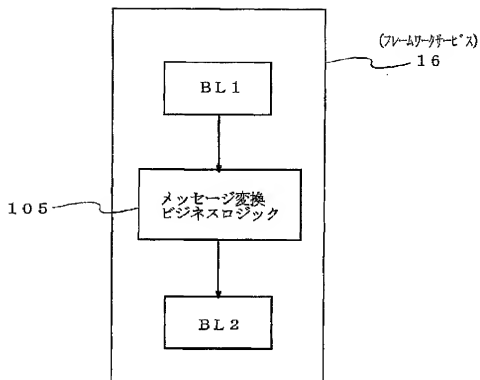


FIG. 24





## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/11532

## A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl.<sup>7</sup> G06F15/00, G06F15/16

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl.<sup>7</sup> G06F15/00, G06F15/16-15/177, G06F13/00, G06F12/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2002
Kokai Jitsuyo Shinan Koho	1971-2002	Toroku Jitsuyo Shinan Koho	1994-2002

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	Shinichi MORISOB, "Application Kan Renkei o Jidoka suru EAI Tool", No.80, Nikkei Business Publications, Inc., 15 November, 1999 (15.11.99), Pages 123 to 127; particularly, page 127, left column, line 4 to right column, line 3; Fig. 2	1-20
Y	Akihiko MATSUMOTO, Tokushu "EAI Solution", Network Computing, Vol.11, No.8, published by Rikku Telecom, 01 August, 1999 (01.08.99), Pages 34 to 47	1,3-7,9-11, 17-19
Y	JP 2000-250768 A (International Business Machines Corp.), 14 September, 2000 (14.09.00), (Family: none)	1,4,10,13,17, 19

☒ Further documents are listed in the continuation of Box C.☐ See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search  
08 April, 2002 (08.04.02)Date of mailing of the international search report  
23 April, 2002 (23.04.02)Name and mailing address of the ISA/  
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/11532

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	"Host Renkel to EAT", Nikkei Computer, No.511, Nikkei Business Publications, Inc., 18 December, 2000 (18.12.00), Pages 251 to 262; particularly, pages 254 to 255	1, 4, 7, 10, 16-20
Y	Tomoo IDA, "Business Rule Chushin no Kaihatsu Shuho", Network Computing, Vol.10, No.9, published by Rikku Telecom, 01 September, 1998 (01.09.98), Pages 55 to 67; particularly, page 60, right column, lines 14 to 35	2
Y	JP 01-253036 A (Nippon Telegraph And Telephone Corp.), 09 October, 1989 (09.10.89), (Family: none)	3, 16, 20
Y	JP 02-113362 A (Nippon Telegraph And Telephone Corp.), 23 April, 1990 (25.04.90), (Family: none)	3
Y	JP 11-15796 A (International Business Machines Corp.), 22 January, 1999 (22.01.99), (Family: none)	1, 7, 10, 13-15, 17-19
Y	EP 1024429 A (Mitsubishi Denki K.K.), 09 November, 1999 (09.11.99), & JP 2000-284980 A	8, 16, 20
Y	JP 01-129345 A (Toshiba Corp.), 22 May, 1989 (22.05.89), (Family: none)	8
Y	JP 02-275563 A (Nippon Telegraph And Telephone Corp.), 09 November, 1990 (09.11.90), (Family: none)	12
Y	JP 03-85663 A (Hitachi, Ltd. and another), 10 April, 1991 (10.04.91), (Family: none)	13-15
Y	JP 62-126457 A (Fujitsu Ltd.), 08 June, 1987 (08.06.87), (Family: none)	13
Y	JP 58-56171 A (Fujitsu Ltd.), 02 April, 1983 (02.04.83), (Family: none)	14, 15

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/11532

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Takao FUKUDA, "J2EE Taio Web Application Server", Nikkei Internet Technology, No.41, Nikkei Business Publications, Inc., 22 November, 2000 (22.11.00), Pages 210 to 219	1-20
A	JP 2000-242697 A (Kokusai Electric Co., Ltd.), 08 September, 2000 (08.09.00), (Family: none)	1-20
A	JP 02-109154 A (Nippon Telegraph And Telephone Corp.), 20 April, 1990 (20.04.90), (Family: none)	1-20
A	JP 62-212763 A (Fujitsu Ltd.), 18 September, 1987 (18.09.87), (Family: none)	1-20
A	JP 59-220866 A (Hitachi Ltd.), 12 December, 1984 (12.12.84), (Family: none)	1-20

A. 発明の属する分野の分類 (国際特許分類 (IPC))	
Int. Cl. 1 G06F15/00, G06F15/16	
B. 調査を行った分野 調査を行った最小限資料 (国際特許分類 (IPC))	
Int. Cl. 1 G06F15/00, G06F15/16-15/177, G06F15/00, G06F12/00	
最小限資料以外の資料で調査を行った分野に含まれるもの 日本国実用新案公報 1992-1996年 日本国公開実用新案公報 1971-2002年 日本国実用新案登録公報 1996-2002年 日本国登録実用新案公報 1994-2002年	
国際調査で利用した電子データベース (データベースの名称、調査に利用した用語)	
C. 関連すると認められる文献	
引用文献の カテゴリー*	引用文献名 及び一部の箇所に関連するときは、その関連する箇所の表示 関連する 請求の範囲の番号
Y	森側真一, アプリケーション関連連携を自動化するEAIツール, 日経オープンシステム, No. 80, 日経B P社発行, 1999.11.15, 第123~127頁, 特に第127頁左欄第4行~右欄第3行及び第2図を参照。 1-20
Y	松本昭彦, 特集 EAIソリューション, ネットワークコンピューティング, Vol. 11, No. 8, 株式会社リックテレコム発行, 1999.08.01, 第34~47頁 1, 3-7, 9-11, 17-19
Y	JP 2000-250768 A (インターナショナル・ビジネス・マシーンス・コーポレーション) 2000.09.14 (ファミリーなし) 1, 4, 10, 13, 17, 19
<input checked="" type="checkbox"/> C欄の続きにも文献が列挙されている。 <input type="checkbox"/> パテントファミリーに関する別紙を参照。	
* 引用文献のカテゴリー 「A」 特に関連のある文献ではなく、一般技術水準を示すもの 「E」 国際出願日以前の出願または特許であるが、国際出願日以後に公表されたもの 「L」 優先権主張に拠る発明の発明又は他の文献の発行日よりしくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」 口頭による開示、使用、展示等に言及する文献 「P」 国際出願日以前で、かつ優先権の主張の基礎となる出願 「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」 同一パテントファミリー文献	
国際調査を完了した日	08.04.02
国際調査機関の名称及びあて先 日本国特許庁 (ISA/J P) 郵便番号 100-8915 東京都千代田区霞が関三丁目4番3号	国際調査報告の発送日 23.04.02 特許庁審査官 (権限のある職員) 坂越 剛史 電話番号 03-3581-1101 内線 3546

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y	ホスト連携とE A I, 日経コンピュータ, No. 511, 日経B P社発行, 2000. 12. 18, 第251~262頁, 特に第254~255頁を参照。	1, 4, 7, 10 , 16-20
Y	依田智夫, ビジネスルール中心の開発手法, ネットワークコンピューティング, Vol. 10, No. 9, 株式会社リックテレコム発行, 1998. 09. 01, 第55~67頁, 特に第60頁右欄第14~35行を参照。	2
Y	JP 01-253036 A(日本電信電話株式会社) 1989. 10. 09 (ファミリーなし)	3, 16, 20
Y	JP 02-113362 A(日本電信電話株式会社) 1990. 04. 25 (ファミリーなし)	3
Y	JP 11-15796 A(インターナショナル・ビジネス・マシーニズ・コーポレーション) 1999. 01. 22 (ファミリーなし)	1, 7, 10, 13-15 , 17-19
Y	EP 1024429 A(MITSUBISHI DENKI KABUSHIKI KAISHA) 1999. 11. 09 & JP 2000-284980 A	8, 16, 20
Y	JP 01-129345 A(株式会社東芝) 1989. 05. 22 (ファミリーなし)	8
Y	JP 02-275563 A(日本電信電話株式会社) 1990. 11. 09 (ファミリーなし)	12
Y	JP 03-85663 A(株式会社日立製作所 外1名) 1991. 04. 10 (ファミリーなし)	13-15
Y	JP 62-126457 A(富士通株式会社) 1987. 06. 08 (ファミリーなし)	13
Y	JP 58-66171 A(富士通株式会社) 1983. 04. 02 (ファミリーなし)	14, 15
A	福田崇男, J 2 E E対応Webアプリケーション・サーバー, 日経インターネットテクノロジー, No. 41, 日経B P社発行, 2000. 11. 22, 第210~219頁	1-20
A	JP 2000-242697 A(国素電気株式会社) 2000. 09. 08 (ファミリーなし)	1-20
A	JP 02-109154 A(日本電信電話株式会社) 1990. 04. 20 (ファミリーなし)	1-20
A	JP 62-212763 A(富士通株式会社) 1987. 09. 18 (ファミリーなし)	1-20
A	JP 59-220866 A(株式会社日立製作所) 1984. 12. 12 (ファミリーなし)	1-20